PROJECT REPORT
ON

INVENTORY MANAGEMENT SYSTEM WITH BARCODE

PAPER NAME: PROJECT-II

PAPER CODE: PROJ-CS781

SEMESTER: VII

of

DEBASISH HALDER	UNIVERSITY ROLL NO. 27600118082
NEHA RAY	UNIVERSITY ROLL NO. 27600119149
JITESH SHAW	UNIVERSITY ROLL NO. 27600119150
ANIRBAN KUMAR MALICK	UNIVERSITY ROLL NO. 27600118093
KANCHAN KUMARI	UNIVERSITY ROLL NO. 27600119155
MUSTAK SK	UNIVERSITY ROLL NO. 27600119126

in

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Accredited by
NATIONAL BOARD OF ACCREDITATION

ACADEMIC SESSION: 2018-2022

BUDGE BUDGE INSTITUTE OF TECHNOLOGY

Nischintapur , Budge Budge, Kolkata-700137

Approved by

All India Council for Technical Education New Delhi, India

Accredited by

National Assessment and Accreditation Council

Affiliated to

Maulana Abul Kalam Azad University of Technology

(Formerly West Bengal University of Technology)
BF-142, Salt Lake, Kolkata-700064

TABLE OF CONTENT

CERTIFICATION

This is to certify that this research work was carried out by :

DEBASISH HALDER	**UNIVERSITY ROLL NO. 27600118082**
NEHA RAY	**UNIVERSITY ROLL NO. 27600119149**
JITESH SHAW	**UNIVERSITY ROLL NO. 27600119150**
ANIRBAN KUMAR MALICK	**UNIVERSITY ROLL NO. 27600118093**
KANCHAN KUMARI	**UNIVERSITY ROLL NO. 27600119155**
MUSTAK SK	**UNIVERSITY ROLL NO. 27600119126**

of the Department of COMPUTER SCIENCE under the supervision of **Prof. Dr. Bimal Dutta (HOD) and Prof. Rupam Sardar**

ACKNOWLEDGMENT

We appreciate our supervisor **Prof. Rupam Sardar** for the supervision and support that he gave, which helped the progression and smoothness of the project.

We extend our thanks to the Entire Computer Science Department of **Budge Budge Institute of Technology** , the H.O.D **Professor Dr. Bimal Datta** and all lecturers who prepared us from the base of computer science.

We would also like to appreciate our friends and special thanks to our parents who encouraged, supported and helped us financially, prayerfully and morally throughout this project.

ABSTRACT

This thesis portrays the Inventory Management System adequately to decide the possibility and convenience of a completed system. The main idea is to trace the trading from the sales registers with extra highlights for deciphering the information. It utilizes a server model with an associated database to permit numerous stores and other locations to be associated. This takes into account later extension while as yet supporting the focus on independent small ventures.

This thesis depicts the IMS adequately to decide the practicality and convenience of a completed framework. The center idea is to follow the offer of things from the sales registers with extra highlights for deciphering the information. It utilizes a customer worker model with an associated information base to permit various stores and distribution centers to be associated. This takes into account later development while as yet supporting the focus on private companies. Inventory Management (IMS) are generally utilized in industry these days to build the effectiveness of thin streams. The conventional technique for IMS included a great deal of administrative work, for example, a bookkeeping page and request list which will be harder to oversee as the capacity becomes greater. In this manner, IMS can be modernized to further expand the effectiveness. This undertaking intended to create IMS programming can store a huge sum of information. Additionally, it has check in/out capacity and search work that runs utilizing examine QR Code. Other than that, the product additionally has the element of low stock notice capacity to caution the client if the thing has low amounts. To build up the product, first the measures of the product are resolved by picking the product advancement devices to build up the product that can accomplish the models of the product. Next the product will be created and fixed. This paper centers on creating IMS programming for SMEs and biomedical field organization use.

INTRODUCTION

➢ What Is Inventory Management?

Inventory management helps companies identify which and how much stock to order at what time. It tracks inventory from purchase to the sale of goods. The practice identifies and responds to trends to ensure there's always enough stock to fulfill customer orders and proper warning of a shortage.

One measurement of good inventory management is inventory turnover. An accounting measurement, inventory turnover reflects how often stock is sold in a period. A business does not want more stock than sales. Poor inventory turnover can lead to deadstock, or unsold stock.

➢ Why Is Inventory Management Important?

Inventory management is vital to a company's health because it helps make sure there is rarely too much or too little stock on hand, limiting the risk of stockouts and inaccurate records.

➢ Benefits of Inventory Management

The two main benefits of inventory management are that it ensures you're able to fulfill incoming or open orders and raises profits. Inventory management also:

- **Saves Money:**

Understanding stock trends means you see how much of and where you have something in stock so you're better able to use the stock you have. This also allows you to keep less stock at each location (store, warehouse), as you're able to pull from anywhere to fulfill orders — all of this decreases costs tied up in inventory and decreases the amount of stock that goes unsold before it's obsolete.

- **Improves Cash Flow:**

With proper inventory management, you spend money on inventory that sells, so cash is always moving through the business.

AIMS AND OBJECTIVES

- **Preventing Dead Stock or Perishability**: With an optimal inventory level, the chances of wastage in the form of goods spoilage or dead stock.
- **Optimizing Storage Cost**: It reduces the chances of maintaining excessive stock, even the requirements are pre-determined, which ultimately cuts done the unnecessary warehousing costs.
- **Maintaining Sufficient Stock**: Now, the production department need not worry about the shortage of raw material or goods because of its constant supply.
- **Enhancing Cash Flow**: Inventory has a significant impact on the cash flow of the company. With effective inventory management, the organization can ensure sufficient liquid cash to enhance its operational efficiency.
- **Reducing the Inventories' Cost Value**: When there is a constant purchase of goods or stock, the organization can ask for discounts and other benefits to decrease the purchase price.

SCOPE AND LIMITATIONS

1. Determination of economic order quantity:
Economic order quantity or economic lot size refers to that number ordered in a single purchase or number of units should be manufactured in a single run, so that the total costs — ordering or set up costs and inventory carrying costs are at the minimum. So, the determination of E.O.Q. is also within the scope of inventory control.

2. Formulation of policy:
The policies of investment procurement, storage, handling, accounting, storages and stock outs, deterioration, obsolescence etc. are to be formulated under the scientific system of inventory control. What, when and how much of purchasing and fixation of minimum and maximum levels is also to be determined for a given period of time.

3. Determination of lead time:
By lead time is meant the time that lapses between the raising of an indent by the stores and the receipt of materials by them. Lead time is of fundamental importance in determining inventory levels.

4. Effectiveness towards running of store:
The deter-mination of policies of the location, layout and materials and storage handling equipments certainly help in the effective working of stores organization.

5. Organization structure:
After determining of inventory policy, the next step is to decide the location, layout and types of storehouse. It facilitates the movement of materials and thus minimize the storage and handling cost of stores.

6. Determination of safety stock:
Safety stock is defined as the difference between the amount stocked to satisfy demand during a certain time interval and the mean expected demand for that period. It is for the purpose of providing protection against depletion. If demand remained constant and lead tin-; is invariable, there would be no fear of shortages and no need for safely stocks.
The exact quantity of safety stock of an item depends upon its lead time, usage value, and variability of lead time demand, carrying charges and the importance of its stock out cost. Again, determination of buffer stock reserve stock is included in the management of inventory.

7. Minimum material handling and storage cost: Stores organization activities are arranged in such a manner that the east of bringing in the store house and issuing from the store house if the various stores, will minimize the storage and materials handling cost of stores.

ANALYSIS OF EXISTING SYSTEM

The existing system for the wholesale services was characterized by manual field registration and file based system which was susceptible to an excessive number of mistakes, data redundancy and the difficulty of updating information about the goods in stock are but just a few of the challenges the company faces.

ANALYSIS OF PROPOSED SYSTEM

Inventory Management System has several functions which enable the admin to do so many jobs efficiently.

- The admin inputs all the current stock and can be able to update if there should be any changes.
- Monitoring all the sales activities can be done by simply printing the report sales.
- Tracking the quantity of goods available in the store, or if need be for a restock. Also

  the admin can be able to generate report on all the activities of the store.

PROPOSED METHODOLOGY

Two development methodologies will be used during the design of this project. They are :-
- Waterfall Development Methodology
- Agile Development Methodology

Approach to Chosen Methology/Methods:-

Waterfall Model Management

If the waterfall model is to be executed properly, each of the phases we outlined earlier must be executed in a linear fashion. Meaning, each phase has to be completed before the next phase can begin, and phases are never repeated—unless there is a massive failure that comes to light in the verification or maintenance phase.

Furthermore, each phase is discrete, and pretty much exists in isolation from stakeholders outside of your team. This is especially true in the requirements phase. Once the customer’s requirements are collected, the customers cease to play any role in the actual development of the software.

Agile Project Management

Agile differs greatly from waterfall in two major ways; namely in regards to linear action and customer involvement. Agile is a nimble and iterative process, where the product is delivered in stages to the customer for them to review and provide feedback.

Instead of having everything planned out by milestones, like in waterfall, Agile operates in “sprints” where prioritized tasks are completed within a short window, typically around two weeks.

These prioritized tasks are fluid, and appear based on the success of previous sprints and customer feedback, rather than having all tasks prioritized at the onset in the requirements phase.

WATERFALL METHODOLOGY APPROACH :-

Requirement and analysis: Amongst all the requirements of the system to be designed are gotten in this stage and written in a requirement specification document. In this stage, inquisition on how the current system is running and the limitations therein was conducted, in order to know where to start the proposed system. Based on the analysis, the basic requirements of the proposed system will then be determined, i.e. the input and output, and elimination of redundancies.

System Design: The necessity determinations from the first phase are concentrated in this phase and the system configuration is prepared. This system configuration helps in determining hardware and system necessities and helps in characterizing the general system engineering.

Integration and Testing: In this stage, as referenced prior, unit testing will be directed. All the units created in the execution stage are coordinated into a system subsequent to testing of every unit. Post integration the whole framework is tried for any issues and faults.

Deployment of system: This will happen after all the previously mentioned stages. When the functional and non-functional testing is done, the item is sent in the customer environment or delivered into the market.

Maintenance: After the deployment, there will be a few issues which will come up in the customer environment. To fix those issues, patches are delivered. Likewise, to upgrade the software product some better forms are delivered. Maintenance is done to convey these modifications to the client environment.

AGILE METHODOLOGY APPROACH :-

Notion: Projects are imagined and organized.

Inception: Team individuals are distinguished, financing is set up, and starting conditions and necessities are talked about. Iteration/Construction: The advancement group attempts to convey working programming software dependent on cycle necessities and feedback.

Delivery: QA (Quality Assurance) testing, inside and outside preparing, documentation advancement, and last arrival of the cycle into creation.

Creation: Ongoing help of the software product.

Retirement: End-of-life exercises, including client notice and relocation

SYSTEM REQUIREMENTS:

1. **HARDWARE REQUIREMENTS**
2. **SOFTWARE REQUIREMENTS**

HARDWARE REQUIREMENTS:

System – PC

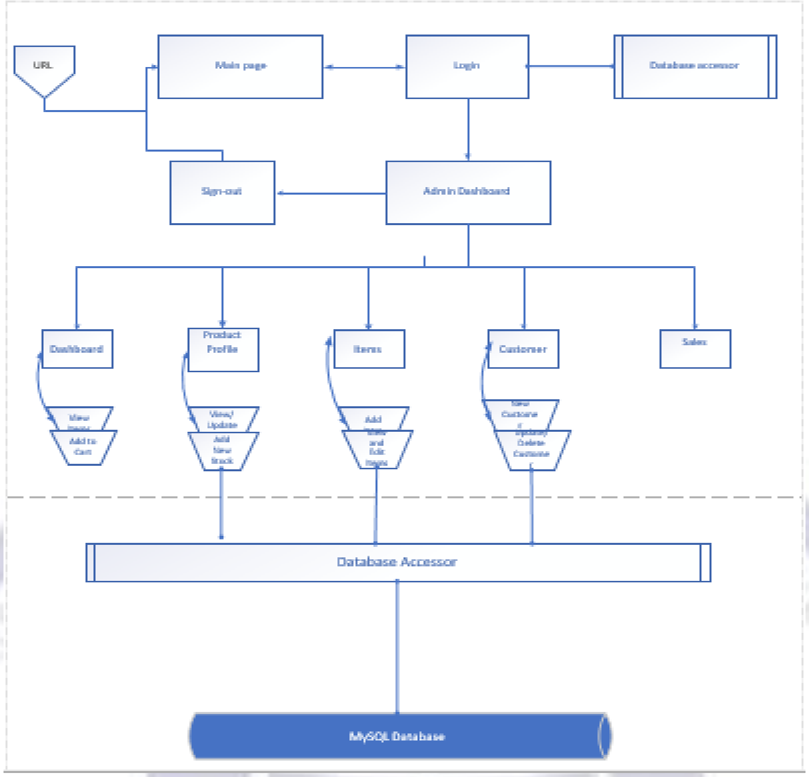
SOFTWARE REQUIREMENTS:

- **OS – WINDOW 10**
- **Visual Studio 2019**
- **MySQLServer Express**
- **C#.NETFramework**

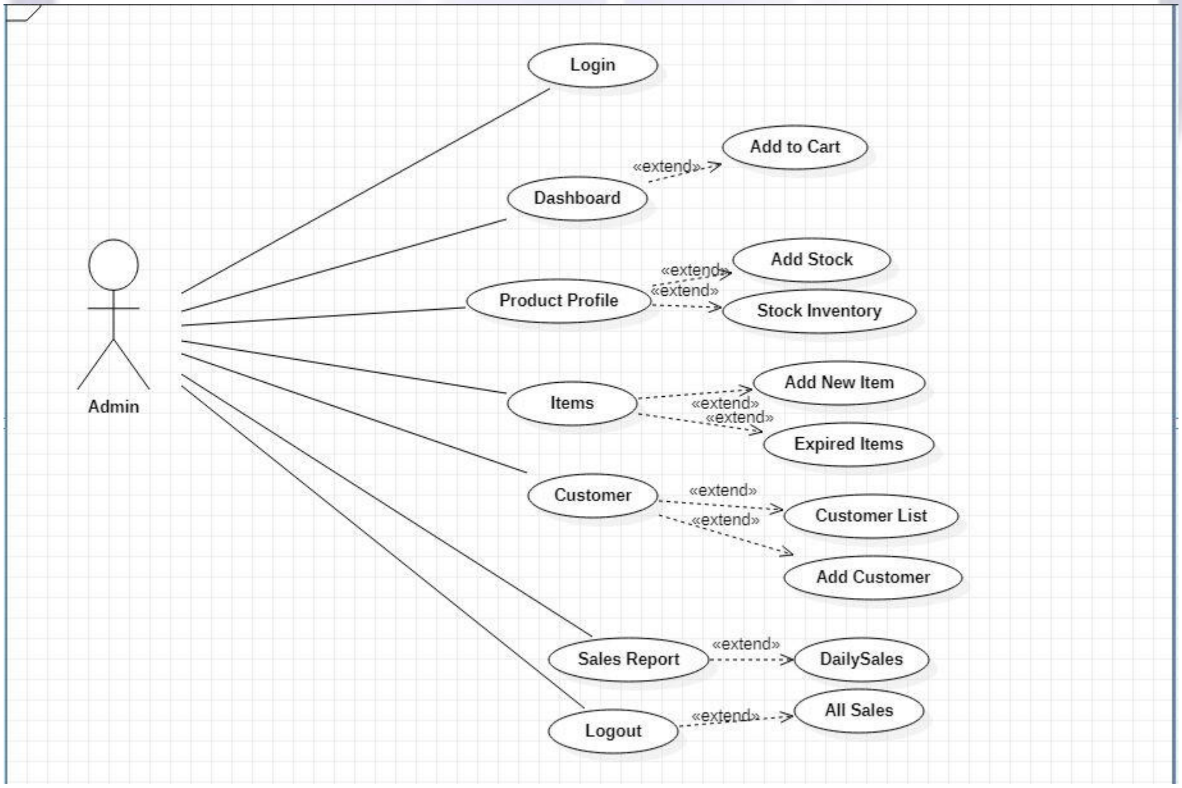
SYSTEM DESIGN

It depicts the systems design and unified modelling language (UML). Several UML diagrams were adopted over the course of the development process such as Data Flow Diagram, Use case Diagram, Application Architecture.

Application Architecture

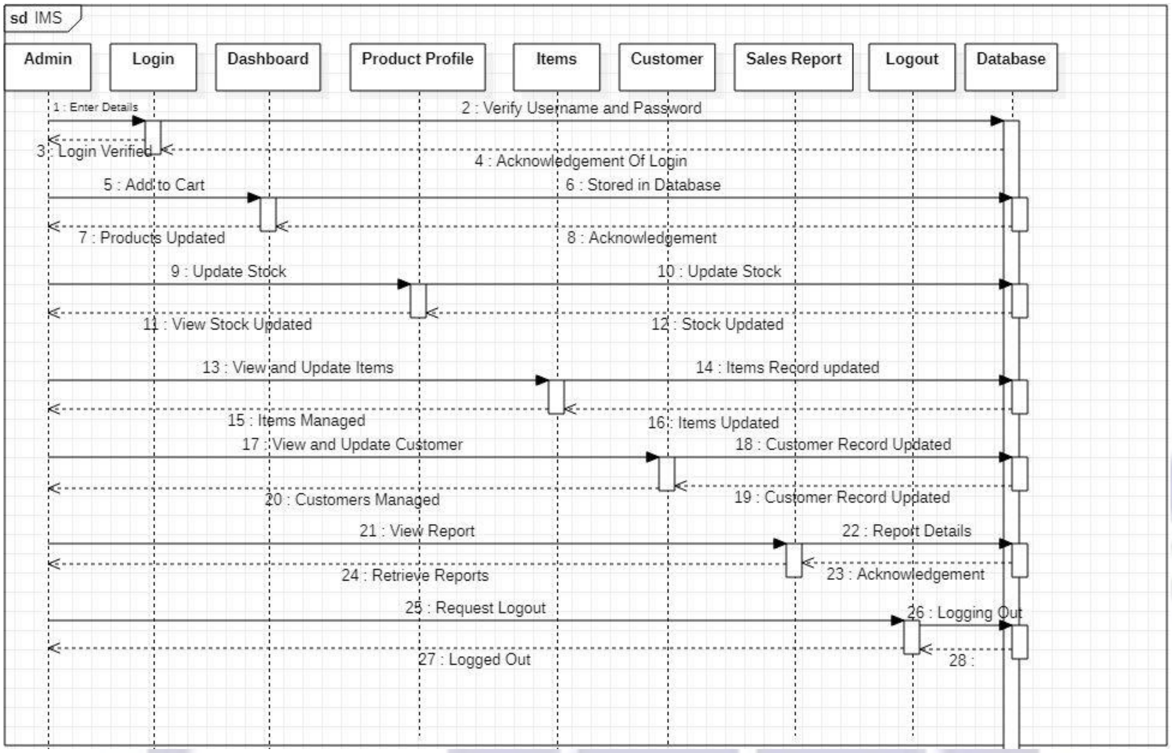


Use Case

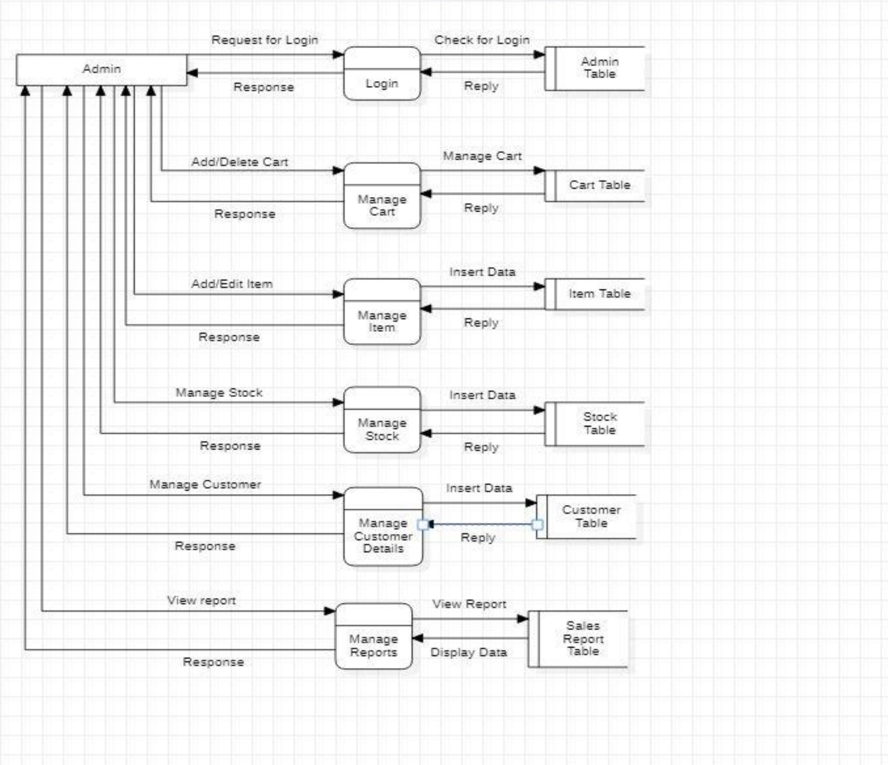


Sequence Diagram

A sequence diagram is a model that shows the process of a task or action from a use case.



Dataflow Diagram



DATABASE DESIGN

CATEGORY TABLE :

FIELD NAME	FIELD TYPE	FIELD LENGTH	DESCRIPTION
Category id	int		Category Id
Category name	varchar	50	Category name

CUSTOMER TABLE :

FIELD NAME	FIELD TYPE	FIELD LENGTH	DESCRIPTION
customerid	int		Customer id
customername	varchar	50	Customer name
customerno	bigint		Customer no
customeraddress	varchar	50	Customer address

ORDER PRODUCT TABLE :

FIELD NAME	FIELD TYPE	FIELD LENGTH	DESCRIPTION
id	int		Order Productid
productid	int		Foreign Key: Id of Stock Table
num	int		Numbering order at the time of order placing
product	varchar	50	Product name
quantity	int		Product quantity
productprpiece	decimal	(18,2)	Product price per piece

totamt	decimal	(18,2)	Total Amount
orderid	int		Foreign Key: Id of Order Table

ORDER TABLE :

NAME	FIELD TYPE	FIELD LENGTH	DESCRIPTION
Orderid	int	50	Order Id
Custid FIELD	int	50	Foreign Key: Id of Customer Table
OrderDate	datetime		Date and time of Order placement
TotalAmt	decimal	(18,2)	Total amount of order
Status	varchar	50	The order is paid or not

STOCK TABLE :

FIELD NAME	FIELD TYPE	FIELD LENGTH	DESCRIPTION
productid	int		Product id
productname	varchar	50	Product name
productquantity	varchar	50	Product quantity
productamount	decimal	(18,2)	Product amount
productcatid	int		Foreign Key: Id of Category Table

SUPPLIER TABLE :

FIELD NAME	FIELD TYPE	FIELD LENGTH	DESCRIPTION
supplierid	int		Supplier id
suppliername	varchar	50	Supplier name
supplierno	bigint		Supplier number
supplieraddress	varchar	50	Supplier address

USER TABLE :

FIELD NAME	FIELD TYPE	FIELD LENGTH	DESCRIPTION
userid	int		User id
username	varchar	50	User name
fullname	varchar	50	Full name
userlevel	varchar	50	User Level
password	varchar	50	Password
telephone	bigint		Telephone number

CHOICE OF PROGRAMMING LANGUAGE

Choosing a programming language depends on your language experience and the scope of the application you are building. While small applications are often created using only one language, it is not uncommon to develop large applications using multiple languages.

The propose application to be built is not a web based application that needs internet facilities to function but a standalone application.

The choice of programming language to use for this program is visual basic. The structure of the Basic programming language is very simple, particularly as to the executable code.

Visual Basic has many new and improved features such as inheritance, interfaces, and overloading that make it a powerful object-oriented programming language. It is particularly easy to develop graphical user interfaces and to connect them to handler functions provided by the application.

Visual Basic fully integrates the .NET Framework and the common language runtime, which together provide language interoperability, garbage collection, enhanced security, and improved versioning support. C# supports single inheritance and creates Microsoft intermediate language (MSIL) as input to native code compilers.

SYSTEM TESTING AND DEBUGGING

Testing is an integral part of software development processes. This is to ensure that the quality requirement of the application is not compromised by testing and debugging program modules before they are integrated, testing the system to ensure an effective inter-operability after integration.

Debugging has to do with fixing of errors encountered during program execution. System testing deals with the real life testing of the system, to ascertain how far it has gone in carrying out the expected task. This was carried out in two phases.

Number one is the source code testing which examine the logic of the program. Secondly, the specification testing which involves the examination of the system as regard to what it should do and how it should be done given specific conditions. This includes inputting data, collecting its output and comparing it with the output of the old system and assessing it to see if it can replace the old system.

DEVELOPMENT

SPLASH FORM :

Code:

```
       using System;
       using System.Collections.Generic;
       using System.ComponentModel;
       using System.Data;
       using System.Drawing;
       using System.Linq;
       using System.Text;
       using System.Threading.Tasks;
       using System.Windows.Forms;

       namespace inventory_management
       {
       public partial class SplashForm : Form
       {
       public SplashForm()
       {
       InitializeComponent();
       }

       int startpoint = 0;

       private void label1_Click(object sender, EventArgs e)
       {

       }

       private void timer1_Tick(object sender, EventArgs e)
       {
       startpoint += 1;
       Progress.Value = startpoint;
       if(Progress.Value ==100)
       {
       Progress.Value = 0;
       timer1.Stop();
       LoginForm login = new LoginForm();
       this.Hide();
       login.Show();
       }

       }

       private void SplashForm_Load(object sender, EventArgs e)
       {
       timer1.Start();
       }
       }
}
```

LOG-IN FORM :

Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
namespace inventory_management
{
public partial class LoginForm : Form
{
public LoginForm()
{
InitializeComponent();
}
SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\DEBASISH\Documents\Inventory_Mana
ge_user.mdf;Integrated Security=True;Connect Timeout=30");
public static string Userlvl = "";
public static string Username = "";
private void label1_Click(object sender, EventArgs e)
{
Application.Exit();
}

private void label2_Click(object sender, EventArgs e)
{

}

private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
Userlvl = "guest";
Dashboard dash = new Dashboard();
dash.Show();
this.Hide();
}

private void checkBox1_CheckedChanged(object sender, EventArgs e)
{

}



private void checkBox1_CheckedChanged_1(object sender, EventArgs e)
{

if (checkBox1.Checked == true)
{
PaswordTb.UseSystemPasswordChar = false;
}
else
{
PaswordTb.UseSystemPasswordChar = true;
}
}

private void linkLabel2_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
usernameTb.Text = "";
PaswordTb.Text = "";
}

private void panel1_Paint(object sender, PaintEventArgs e)
{

}

private void button1_Click(object sender, EventArgs e)
{

}

private void usernameTb_TextChanged(object sender, EventArgs e)
{

}

private void guna2Button1_Click(object sender, EventArgs e)
{
Userlvl = UserlevelTb.SelectedItem.ToString();
Username = usernameTb.Text;
con.Open();
string pass = PaswordTb.Text;
SqlDataAdapter sda = new SqlDataAdapter("Select Count(*) from UserTbl where username ='"
+ usernameTb.Text + "' and password = '" + (pass.GetHashCode()).GetHashCode() + "' and
userlevel='"+UserlevelTb.SelectedItem.ToString()+"'", con);
DataTable dt = new DataTable();
sda.Fill(dt);
if(dt.Rows[0][0].ToString() == "1")
{
Dashboard dash = new Dashboard();
dash.Show();
this.Hide();
}
else
{
MessageBox.Show("Wrong UserName Or Password");
}
con.Close();
}

private void label3_Click(object sender, EventArgs e)
{

}

private void LoginForm_Load(object sender, EventArgs e)
{
        }

        private void PaswordTb_TextChanged(object sender, EventArgs e)
        {

        }

        private void UserlevelTb_SelectedIndexChanged(object sender, EventArgs e)
        {

        }
        }
}
```

DASHBOARD :

Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace inventory_management
{
    public partial class Dashboard : Form
    {
        public Dashboard()
        {
            InitializeComponent();
        }

        private void Dashboard_Load(object sender, EventArgs e)
        {
            label10.Text = LoginForm.Userlvl;
        }

        private void guna2PictureBox6_Click(object sender, EventArgs e)
        {
            System.Threading.Thread.Sleep(500);
            Category cw = new Category();
            cw.Show();
            this.Hide();
        }

        private void guna2PictureBox5_Click(object sender, EventArgs e)
        {
            if(LoginForm.Userlvl == "guest")
            {
                MessageBox.Show("You are not authorise to access this option");
            }
            else
            {
                Customers cw = new Customers();
                cw.Show();
                this.Hide();
            }
        }

        private void guna2PictureBox2_Click(object sender, EventArgs e)
        {
            if (LoginForm.Userlvl == "guest")
            {
                MessageBox.Show("You are not authorise to access this option");
            }
            else
            {
                ManageUser cw = new ManageUser();
                cw.Show();
                this.Hide();
            }
        }

        private void guna2PictureBox3_Click(object sender, EventArgs e)
        {
            Product cw = new Product();
            cw.Show();
            this.Hide();
        }

        private void guna2PictureBox4_Click(object sender, EventArgs e)
        {
            if (LoginForm.Userlvl == "guest")
            {
                MessageBox.Show("You are not authorise to access this option");
            }
            else
            {
                Order cw = new Order();
                cw.Show();
                this.Hide();
            }
        }

        private void guna2PictureBox1_Click(object sender, EventArgs e)
        {
            if (LoginForm.Userlvl == "guest")
            {
                MessageBox.Show("You are not authorise to access this option");
            }
            else
            {
                supplier cw = new supplier();
                cw.Show();
                this.Hide();
            }
        }

        private void label2_Click(object sender, EventArgs e)
        {
            Category cw = new Category();
            cw.Show();
            this.Hide();
        }
        private void label3_Click(object sender, EventArgs e)
        {
            if (LoginForm.Userlvl == "guest")
            {
                MessageBox.Show("You are not authorise to access this option");
            }
            else
            {
                Customers cw = new Customers();
                cw.Show();
                this.Hide();
            }
        }

        private void label5_Click(object sender, EventArgs e)
        {
            if (LoginForm.Userlvl == "guest")
            {
                MessageBox.Show("You are not authorise to access this option");
            }
            else
            {
                ManageUser cw = new ManageUser();
                cw.Show();
                this.Hide();
            }
        }

        private void label6_Click(object sender, EventArgs e)
        {
            Product cw = new Product();
            cw.Show();
            this.Hide();
        }

        private void label7_Click(object sender, EventArgs e)
        {
            if (LoginForm.Userlvl == "guest")
            {
                MessageBox.Show("You are not authorise to access this option");
            }
            else
            {
                Order cw = new Order();
                cw.Show();
                this.Hide();
            }
        }

        private void label8_Click(object sender, EventArgs e)
        {
            if (LoginForm.Userlvl == "guest")
            {
                MessageBox.Show("You are not authorise to access this option");
            }
            else
            {
                supplier cw = new supplier();
                cw.Show();
                this.Hide();
            }
        }

        private void label4_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void guna2GradientButton1_Click(object sender, EventArgs e)
        {
            LoginForm lg = new LoginForm();
            lg.Show();
            this.Hide();
        }

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void label10_Click(object sender, EventArgs e)
        {

        }

        private void label9_Click(object sender, EventArgs e)
        {

        }
    }
}
```

MANAGE CUSTOMERS :

Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace inventory_management
{
    public partial class Customers : Form
    {
        public Customers()
        {
            InitializeComponent();
        }
        SqlConnection con = new SqlConnection(@"Data 
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\DEBASISH\Documents\Inventory_Mana
ge_user.mdf;Integrated Security=True;Connect Timeout=30");

        private void label4_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        void populate()
        {
            try
            {
                con.Open();
                string MYquery = "select customerid, 
customername,customerno,customeraddress from CustomerTbl";
                SqlDataAdapter da = new SqlDataAdapter(MYquery, con);
                SqlCommandBuilder builder = new SqlCommandBuilder(da);
                var ds = new DataSet();
                da.Fill(ds);
                datagridviewmc.DataSource = ds.Tables[0];
                con.Close();
            }
            catch (Exception E)
            {
                MessageBox.Show(E.Message);
                con.Close();
            }
        }
        private void BackToHomeLinkMc_LinkClicked(object sender, 
LinkLabelLinkClickedEventArgs e)
        {
            Dashboard home = new Dashboard();
            home.Show();
            this.Hide();
        }

        private void AddButtonMc_Click(object sender, EventArgs e)
        {
            try
            {
                if (  CustomerNameMc.Text == "" || CustomerPhoneNoMc.Text == "" || 
CustomerAddressMc.Text == "")
                {
                    MessageBox.Show(" Fill all the fields");
                }
                else
                {
                    con.Open();
                    SqlCommand cmd = new SqlCommand("insert into 
CustomerTbl(customername, customerno, customeraddress) values('" + CustomerNameMc.Text + 
"','" + CustomerPhoneNoMc.Text + "','" + CustomerAddressMc.Text + "')", con);
                    cmd.ExecuteNonQuery();
                    MessageBox.Show("Customer Successfully Added");
                    con.Close();
                }
                populate();
            }
            catch (Exception E)
            {
                MessageBox.Show(E.Message);
                MessageBox.Show("Please ensure if Customer Name and Customer phone number 
are unique!");
                con.Close();
            }
        }

        private void Customers_Load(object sender, EventArgs e)
        {
            populate();
            label10.Text = LoginForm.Userlvl;
        }

        private void DeleteButtonMc_Click(object sender, EventArgs e)
        {
            if (CustomerNameMc.Text == "")
            {
                MessageBox.Show("Please Select a Customer!");

            }
            else
            {
                con.Open();
                string myquery = "delete from CustomerTbl where customerid='" + 
datagridviewmc.SelectedRows[0].Cells[0].Value.ToString() + "'; ";
                SqlCommand cmd = new SqlCommand(myquery, con);
                cmd.ExecuteNonQuery();
                MessageBox.Show("Customer Successfully Deleted");
                con.Close();
                populate();
            }
        }

        private void EditButtonMc_Click(object sender, EventArgs e)
        {
            try
            {
                con.Open();
                SqlCommand cmd = new SqlCommand("update CustomerTbl set customername='" + 
CustomerNameMc.Text + "',customerno='" + CustomerPhoneNoMc.Text + "',customeraddress='" + 
CustomerAddressMc.Text + "' where customerid = '" + 
datagridviewmc.SelectedRows[0].Cells[0].Value.ToString() + "';" ,con);
                cmd.ExecuteNonQuery();
                MessageBox.Show("Customer Successfully Updated");
                con.Close();
                populate();
            }
            catch(Exception E)
            {
                MessageBox.Show(E.Message);
                MessageBox.Show("Please ensure if Customer Name and Customer phone number 
are unique!");
                con.Close();
            }


        }

        private void datagridviewmc_CellContentClick(object sender, 
DataGridViewCellEventArgs e)
```

```
        {
            CustomerNameMc.Text = 
datagridviewmc.SelectedRows[0].Cells[1].Value.ToString();
            CustomerPhoneNoMc.Text = 
datagridviewmc.SelectedRows[0].Cells[2].Value.ToString();
            CustomerAddressMc.Text = 
datagridviewmc.SelectedRows[0].Cells[3].Value.ToString();
            con.Open();
            SqlDataAdapter sda = new SqlDataAdapter("select Count(*) from OrderTbl where 
Custid = '" + datagridviewmc.SelectedRows[0].Cells[0].Value.ToString() + "' ", con);
            DataTable dt = new DataTable();
            sda.Fill(dt);
            OrderLabel.Text = dt.Rows[0][0].ToString();

            SqlDataAdapter sda1 = new SqlDataAdapter("select Sum(TotalAmt) from OrderTbl 
where Custid = '" + datagridviewmc.SelectedRows[0].Cells[0].Value.ToString() + "' ", 
con);
            DataTable dt1 = new DataTable();
            sda1.Fill(dt1);
            AmountLabel.Text = dt1.Rows[0][0].ToString();

            SqlDataAdapter sda2 = new SqlDataAdapter("select Max(OrderDate) from OrderTbl 
where Custid = '" + datagridviewmc.SelectedRows[0].Cells[0].Value.ToString() + "' ", 
con);
            DataTable dt2 = new DataTable();
            sda2.Fill(dt2);
            DateLabel.Text = dt2.Rows[0][0].ToString();
            con.Close();

        }

        private void panel9_Paint(object sender, PaintEventArgs e)
        {

        }

        private void panel1_Paint(object sender, PaintEventArgs e)
        {

        }

        private void guna2Button3_Click(object sender, EventArgs e)
        {
            if (datagridviewmc.Rows.Count > 0)
            {
                Microsoft.Office.Interop.Excel.Application xcelApp = new 
Microsoft.Office.Interop.Excel.Application();
                xcelApp.Application.Workbooks.Add(Type.Missing);
                for (int i = 1; i < datagridviewmc.Columns.Count + 1; i++)
                {
                    xcelApp.Cells[1, i] = datagridviewmc.Columns[i - 1].HeaderText;
                }
                for (int i = 0; i < datagridviewmc.Rows.Count; i++)
                {
                    for (int j = 0; j < datagridviewmc.Columns.Count; j++)
                    {
                        xcelApp.Cells[i + 2, j + 1] = 
datagridviewmc.Rows[i].Cells[j].Value.ToString();
                    }
                }
                xcelApp.Columns.AutoFit();
                xcelApp.Visible = true;
            }
        }
    }
}
```

MANAGE CATEGORIES :

Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace inventory_management
{
    public partial class Category : Form
    {
        public Category()
        {
            InitializeComponent();
        }
        SqlConnection con = new SqlConnection(@"Data 
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\DEBASISH\Documents\Inventory_Mana
ge_user.mdf;Integrated Security=True;Connect Timeout=30");

        private void label4_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
        void populate()
        {
            try
            {
                con.Open();
                string MYquery = "select * from CategoryTbl";
                SqlDataAdapter da = new SqlDataAdapter(MYquery, con);
                SqlCommandBuilder builder = new SqlCommandBuilder(da);
                var ds = new DataSet();
                da.Fill(ds);
                datagridviewMC.DataSource = ds.Tables[0];
                con.Close();
            }
            catch (Exception E)
            {
                MessageBox.Show(E.Message);
                con.Close();
            }
        }
        private void guna2TextBox1_TextChanged(object sender, EventArgs e)
        {

        }

        private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs
e)
        {

            Dashboard home = new Dashboard();
            home.Show();
            this.Hide();
        }

        private void Category_Load(object sender, EventArgs e)
        {
            populate();
            label10.Text = LoginForm.Userlvl;
        }

        private void guna2Button1_Click(object sender, EventArgs e)
        {
            if (LoginForm.Userlvl == "guest")
            {
                MessageBox.Show("You are not authorised to make changes");
            }
            else
            {
                try
                {
                    if ( CategorynameMC.Text == "")
                    {
                        MessageBox.Show(" Fill all the fields");
                    }
                    else
                    {
                        con.Open();
                        SqlCommand cmd = new SqlCommand("insert into
CategoryTbl(categoryname) values('" + CategorynameMC.Text + "')", con);
                        cmd.ExecuteNonQuery();
                        MessageBox.Show("Category Successfully Added");
                        con.Close();
                    }
                    populate();
                }
                catch (Exception E)
                {
                    MessageBox.Show(E.Message);
                    MessageBox.Show("Please ensure if the Category has an Unique name");
                    con.Close();

                }
            }
        }

        private void guna2Button2_Click(object sender, EventArgs e)
        {
            if (LoginForm.Userlvl == "guest")
            {
                MessageBox.Show("You are not authorised to make changes");
            }
           else if (CategorynameMC.Text == "")
            {
                MessageBox.Show("Please Select a Category!");

            }
            else
            {
                con.Open();
                string myquery = "delete from CategoryTbl where categoryid='" +
datagridviewMC.SelectedRows[0].Cells[0].Value.ToString() + "'; ";
                SqlCommand cmd = new SqlCommand(myquery, con);
                cmd.ExecuteNonQuery();
                MessageBox.Show("Category Successfully Deleted");
                con.Close();
                populate();
            }
        }

        private void datagridviewMC_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {
            CategorynameMC.Text =
datagridviewMC.SelectedRows[0].Cells[1].Value.ToString();
        }

        private void guna2Button3_Click(object sender, EventArgs e)
        {
            if(datagridviewMC.Rows.Count > 0)
            {
                Microsoft.Office.Interop.Excel.Application xcelApp = new
Microsoft.Office.Interop.Excel.Application();
                xcelApp.Application.Workbooks.Add(Type.Missing);
                for(int i = 1; i < datagridviewMC.Columns.Count +1; i++)
                {
                    xcelApp.Cells[1, i] = datagridviewMC.Columns[i - 1].HeaderText;
                }
                for (int i = 0; i < datagridviewMC.Rows.Count ; i++)
                {
                    for(int j = 0; j < datagridviewMC.Columns.Count; j++)
                    {
                        xcelApp.Cells[i + 2, j + 1] =
datagridviewMC.Rows[i].Cells[j].Value.ToString();
                    }
                }
                xcelApp.Columns.AutoFit();
                xcelApp.Visible = true;
            }
        }
    }
}
```

MANAGE USERS :

Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
namespace inventory_management
{
    public partial class ManageUser : Form
    {
        public ManageUser()
        {
            InitializeComponent();
        }
        SqlConnection con = new SqlConnection(@"Data 
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\DEBASISH\Documents\Inventory_Manage_user.m
df;Integrated Security=True;Connect Timeout=30");
        private void panel1_Paint(object sender, PaintEventArgs e)
        {

        }
        void populate()
        {
            try
            {
                con.Open();
                if(LoginForm.Userlvl == "admin")
                {
                    string MYquery = "select userid,username,userlevel,telephone from UserTbl";
                    SqlDataAdapter da = new SqlDataAdapter(MYquery, con);
                    SqlCommandBuilder builder = new SqlCommandBuilder(da);
                    var ds = new DataSet();
                    da.Fill(ds);
                    dataGridViewmu.DataSource = ds.Tables[0];
                    Userlevelmu.Visible = true;
                }
                else
                {
                    string MYquery = "select userid,username,userlevel,telephone from UserTbl 
where username = '"+ LoginForm.Username +"'";
                    SqlDataAdapter da = new SqlDataAdapter(MYquery, con);
                    SqlCommandBuilder builder = new SqlCommandBuilder(da);
                    var ds = new DataSet();
                    da.Fill(ds);
                    dataGridViewmu.DataSource = ds.Tables[0];
                    Usernamemu.ReadOnly = true;
                    Userlevelmu.Visible = false;
                }

                con.Close();
            }
            catch(Exception E)
            {
                MessageBox.Show(E.Message);
            }
        }
        private void label2_Click(object sender, EventArgs e)
        {

        }

        private void ManageUser_Load(object sender, EventArgs e)
        {
            populate();
            label10.Text = LoginForm.Userlvl;
        }
        private void passwordMu_TextChanged(object sender, EventArgs e)
        {

        }

        private void usernameMu_TextChanged(object sender, EventArgs e)
        {

        }

        private void label3_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
        {

            Dashboard home = new Dashboard();
            home.Show();
            this.Hide();
        }

        private void guna2TextBox1_TextChanged(object sender, EventArgs e)
        {

        }

        private void guna2TextBox2_TextChanged(object sender, EventArgs e)
        {

        }

        private void guna2Button1_Click(object sender, EventArgs e)
        {
            if(LoginForm.Userlvl == "staff")
            {
                MessageBox.Show("You are not authorise to add user ");
            }
            else
            {
                try
                {
                    if (Usernamemu.Text == "" || Fullnamemu.Text == "" ||
Userlevelmu.SelectedItem.ToString() == "" || passwordMu.Text == "" || Telephonemu.Text == "")
                    {
                        MessageBox.Show(" Fill all the fields");
                    }
                    else
                    {
                        con.Open();
                        string pass = passwordMu.Text;
                        SqlCommand cmd = new SqlCommand("insert into UserTbl values('" +
Usernamemu.Text + "','" + Fullnamemu.Text + "','" + Userlevelmu.SelectedItem.ToString() + "','" +
(pass.GetHashCode()).GetHashCode() + "','" + Telephonemu.Text + "')", con);
                        cmd.ExecuteNonQuery();
                        MessageBox.Show("User Successfully Added");
                        con.Close();
                    }
                    populate();
                }
                catch (Exception E)
                {
                    MessageBox.Show(E.Message);
                    MessageBox.Show("Please ensure the User Id and Telephone no. are unique!");
                    con.Close();

                }
            }

        }

        private void dataGridViewmu_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {

        }

        private void guna2Button2_Click(object sender, EventArgs e)
        {
            if(LoginForm.Userlvl == "staff")
            {
                MessageBox.Show("You are not authorised to delete your own Account.\n Please ask
the admin to do so.");
            }
            else
            {
                try
                {


                    if (Usernamemu.Text == "")
                    {
                        MessageBox.Show("Enter select the user ");

                    }
                    else
                    {
                        con.Open();
                        string myquery = "delete from UserTbl where userid='" +
dataGridViewmu.SelectedRows[0].Cells[0].Value.ToString() + "'; ";
                        SqlCommand cmd = new SqlCommand(myquery, con);
                        cmd.ExecuteNonQuery();
                        MessageBox.Show("User Successfully Deleted");
                        con.Close();
                        populate();
                    }
                }
                catch (Exception E)
                {
                    MessageBox.Show(E.Message);
                    con.Close();
                }
            }

        }

        private void dataGridViewmu_CellContentClick_1(object sender, DataGridViewCellEventArgs e)
        {

            Usernamemu.Text = dataGridViewmu.SelectedRows[0].Cells[1].Value.ToString();
            Userlevelmu.SelectedItem = dataGridViewmu.SelectedRows[0].Cells[2].Value.ToString();
            Telephonemu.Text = dataGridViewmu.SelectedRows[0].Cells[3].Value.ToString();
            Fullnamemu.Text = "";
            passwordMu.Text = "";
        }

        private void guna2Button3_Click(object sender, EventArgs e)
        {
            try
            {
                con.Open();
                if (Fullnamemu.Text == "" && passwordMu.Text == "")
                {
                    string pass = "1234";
                    SqlCommand cmd = new SqlCommand("update UserTbl set 
username='"+Usernamemu.Text+"',fullname='"+Usernamemu.Text+"',userlevel='"+Userlevelmu.SelectedIte
m.ToString()+"',password='" +(pass.GetHashCode()).GetHashCode()+"', 
telephone='"+Telephonemu.Text+"' where userid='"+ 
dataGridViewmu.SelectedRows[0].Cells[0].Value.ToString() + "'", con);
                    cmd.ExecuteNonQuery();
                    MessageBox.Show("User Successfully Updated");

                }
                else if(Fullnamemu.Text == "" && passwordMu.Text != "")
                {
                    string pass = passwordMu.Text;
                    SqlCommand cmd = new SqlCommand("update UserTbl set username='" + 
Usernamemu.Text + "',fullname='" + Usernamemu.Text + "',userlevel='" + 
Userlevelmu.SelectedItem.ToString() + "',password='" + (pass.GetHashCode()).GetHashCode() + "', 
telephone='" + Telephonemu.Text + "' where userid='" + 
dataGridViewmu.SelectedRows[0].Cells[0].Value.ToString() + "'", con);
                    cmd.ExecuteNonQuery();
                    MessageBox.Show("User Successfully Updated");
                }
                else if(passwordMu.Text =="" && Fullnamemu.Text != "")
                {
                    string pass = "1234";
                    SqlCommand cmd = new SqlCommand("update UserTbl set username='" + 
Usernamemu.Text + "',fullname='" + Fullnamemu.Text + "',userlevel='" + 
Userlevelmu.SelectedItem.ToString() + "',password='" + (pass.GetHashCode()).GetHashCode() + "', 
telephone='" + Telephonemu.Text + "' where userid='" + 
dataGridViewmu.SelectedRows[0].Cells[0].Value.ToString() + "'", con);
                    cmd.ExecuteNonQuery();
                    MessageBox.Show("User Successfully Updated");
                }
                else
                {
                    string pass = passwordMu.Text;
                    SqlCommand cmd = new SqlCommand("update UserTbl set username='" +
Usernamemu.Text + "',fullname='" + Fullnamemu.Text + "',userlevel='" +
Userlevelmu.SelectedItem.ToString() + "',password='" + (pass.GetHashCode()).GetHashCode() + "',
telephone='" + Telephonemu.Text + "' where userid='" +
dataGridViewmu.SelectedRows[0].Cells[0].Value.ToString() + "'", con);
                    cmd.ExecuteNonQuery();
                    MessageBox.Show("User Successfully Updated");
                }
                con.Close();
                populate();
            }
            catch(Exception E)
            {
                MessageBox.Show(E.Message);
                MessageBox.Show("Please ensure the user has an unique name and telephone");
                con.Close();
            }
        }

        private void Telephonemu_TextChanged(object sender, EventArgs e)
        {

        }

        private void guna2Button4_Click(object sender, EventArgs e)
        {
            if (dataGridViewmu.Rows.Count > 0)
            {
                Microsoft.Office.Interop.Excel.Application xcelApp = new
Microsoft.Office.Interop.Excel.Application();
                xcelApp.Application.Workbooks.Add(Type.Missing);
                for (int i = 1; i < dataGridViewmu.Columns.Count + 1; i++)
                {
                    xcelApp.Cells[1, i] = dataGridViewmu.Columns[i - 1].HeaderText;
                }
                for (int i = 0; i < dataGridViewmu.Rows.Count; i++)
                {
                    for (int j = 0; j < dataGridViewmu.Columns.Count; j++)
                    {
                        xcelApp.Cells[i + 2, j + 1] =
dataGridViewmu.Rows[i].Cells[j].Value.ToString();
                    }
                }
                xcelApp.Columns.AutoFit();
                xcelApp.Visible = true;
            }
        }
    }
}
```

MANAGE STOCK :

Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.IO;
using System.Diagnostics;


namespace inventory_management
{
    public partial class Product : Form
    {
        public Product()
        {
            InitializeComponent();
        }
        SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\DEBASISH\Documents\Inventory_Manage_user.mdf;Integrated
Security=True;Connect Timeout=30");
        void Fillcombo()
        {

            string query = "select * from CategoryTbl;";
            SqlCommand cmd = new SqlCommand(query, con);
            SqlDataReader rdr;
            try
            {
                con.Open();
                DataTable dt = new DataTable();
                dt.Columns.Add("categoryname", typeof(string));
                rdr = cmd.ExecuteReader();
                dt.Load(rdr);
                CategoryMS.ValueMember = "categoryname";
                CategoryMS.DataSource = dt;
                SearchcomboMS.ValueMember = "categoryname";
                SearchcomboMS.DataSource = dt;
                con.Close();
            }
            catch(Exception E)
            {
                MessageBox.Show(E.Message);
                con.Close();
            }
        }

        void populate()
        {
            try
            {
                con.Open();
                string MYquery = "select productid, productname, productquantity, productamount from StockTbl";
                SqlDataAdapter da = new SqlDataAdapter(MYquery, con);
                SqlCommandBuilder builder = new SqlCommandBuilder(da);
                var ds = new DataSet();
                da.Fill(ds);
                datagridviewMS.DataSource = ds.Tables[0];
                con.Close();
            }
            catch (Exception E)
            {
                MessageBox.Show(E.Message);
                con.Close();
            }
        }
        void filterbycategory()
        {
            try
            {
                con.Open();
                int combocat;
                int myint = 0;
                string query1 = "select * from CategoryTbl where categoryname='" + CategoryMS.SelectedValue.ToString() + "';";
                SqlCommand cmd1 = new SqlCommand(query1, con);
                SqlDataReader rdr = null;

                rdr = cmd1.ExecuteReader();
                while (rdr.Read())
                {
                    myint = rdr.GetInt32(0);
                }
                rdr.Close();
                combocat = myint;
                string MYquery = "select productid, productname, productquantity, productamount from StockTbl where productcatid= '" +
combocat + "' ";
                SqlDataAdapter da = new SqlDataAdapter(MYquery, con);
                SqlCommandBuilder builder = new SqlCommandBuilder(da);
                var ds = new DataSet();
                da.Fill(ds);
                datagridviewMS.DataSource = ds.Tables[0];
                con.Close();
            }
            catch (Exception E)
            {
                MessageBox.Show(E.Message);
                con.Close();
            }
        }
        private void label4_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }


        private void guna2Button1_Click(object sender, EventArgs e)
        {
            if (LoginForm.Userlvl == "guest")
            {
                MessageBox.Show("You are not authorised to make changes");
            }
            else
            {
                try
                {
                    if ( ProductnameMS.Text == "" || ProductquantityMS.Text == "" || ProductamountMS.Text == "" ||
CategoryMS.SelectedValue.ToString() == "")
                    {
                        MessageBox.Show(" Fill all the fields");
                    }
                    else
                    {
                        con.Open();
                        int combocat;
                        int myint = 0;
                        string query1 = "select * from CategoryTbl where categoryname='"+CategoryMS.SelectedValue.ToString()+"';";
                        SqlCommand cmd1 = new SqlCommand(query1, con);
                        SqlDataReader rdr;

                        rdr = cmd1.ExecuteReader();
                        while(rdr.Read())
                        {
                            myint = rdr.GetInt32(0);
                        }
                        rdr.Close();
                        combocat = myint;
                        //MessageBox.Show(rdr.GetInt32(0).ToString());
                        //combocat = rdr.GetInt32(0);

                        SqlCommand cmd = new SqlCommand("insert into StockTbl(productname, productquantity, productamount,
productcatid) values('" + ProductnameMS.Text + "','" + ProductquantityMS.Text + "','" + ProductamountMS.Text + "','" + combocat +
"')", con);
                        cmd.ExecuteNonQuery();
                        MessageBox.Show("Product Successfully Added");
                        con.Close();
                        populate();


                        int i = 1;
                        while(i<= Convert.ToInt32(ProductquantityMS.Text))
                        {
                            string prn1 = @"
^Q100,3
^W100
^H5
^P1
^S2
^AD
^C1
^R0
~Q+0
^O0
^D0
^E12
~R255
^XSET,ROTATION,0
^L
Dy2-me-dd
Th:m:s
AD,132,36,2,2,0,0E,ARS Technologies
AD,70,240,1,1,0,0E,Product Price
AD,64,148,1,1,0,0E,Product Id
AD,66,194,1,1,0,0E,Product Name
AD,350,244,1,1,0,0E,";
                    string prn2 = @"
AD,344,152,1,1,0,0E,";
                    string prn3 = @"
AD,346,198,1,1,0,0E,";
                    string prn4 = @"
BQ,76,316,8,20,80,0,1,";
                    string prn5 = @"
E";



                    string prodid = Convert.ToString( datagridviewMS.Rows[datagridviewMS.RowCount-1].Cells[0].Value);



                    string barcode = (prodid).PadLeft(4, '0');
                    string finalPRN = prn1 + ProductamountMS.Text + prn2 + prodid + prn3 + ProductnameMS.Text + prn4 + barcode
+ prn5;
                    StreamWriter sw = new StreamWriter("C:\\Users\\DEBASISH\\AppData\\Roaming\\Godex\\Invent" +
                        "oryManagementPRN.txt");
                    sw.Write(finalPRN);
                    sw.Close();


                    var processStartInfo = new ProcessStartInfo();
                    processStartInfo.WorkingDirectory = @"C:\\Users\\DEBASISH\\AppData\\Roaming\\Microsoft\\Windows\\Start
Menu\\Programs\\System Tools";
                    Process process1 = new Process();
                    process1.StartInfo.FileName = "cmd.exe";
                    process1.StartInfo.Arguments = "/C Type
C:\\Users\\DEBASISH\\AppData\\Roaming\\Godex\\InventoryManagementPRN.txt>\\\\LAPTOP-7EDS20EI\\Godex";
                    Process proc = Process.Start(process1.StartInfo);
                    i = i + 1;
                }


            }
            populate();
        }
        catch (Exception E)
        {
            MessageBox.Show(E.Message);
            MessageBox.Show("Please ensure the product has an unique name");
            con.Close();

        }
    }
}
```

```
        private void textBox2_TextChanged(object sender, EventArgs e)
        {

        }

        private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
        {

            Dashboard home = new Dashboard();
            home.Show();
            this.Hide();
        }

        private void Product_Load(object sender, EventArgs e)
        {
            Fillcombo();
            populate();
            label10.Text = LoginForm.Userlvl;
        }

        private void CategoryMS_SelectedIndexChanged(object sender, EventArgs e)
        {

        }

        private void guna2Button2_Click(object sender, EventArgs e)
        {
            if (LoginForm.Userlvl == "guest")
            {
                MessageBox.Show("You are not authorised to make changes");
            }
            else
            {
                try
                {


                    if (ProductnameMS.Text == "")
                    {
                        MessageBox.Show("Please Select a product");

                    }
                    else
                    {
                        con.Open();
                        string myquery = "delete from StockTbl where productid='" +
datagridviewMS.SelectedRows[0].Cells[0].Value.ToString() + "'; ";
                        SqlCommand cmd = new SqlCommand(myquery, con);
                        cmd.ExecuteNonQuery();
                        MessageBox.Show("Product Successfully Deleted");
                        con.Close();
                        populate();
                    }
                }
                catch (Exception E)
                {
                    MessageBox.Show(E.Message);
                    MessageBox.Show("Please ensure the product has an unique name");
                    con.Close();
                }
            }
        }

        private void datagridviewMS_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {
            ProductnameMS.Text = datagridviewMS.SelectedRows[0].Cells[1].Value.ToString();
            ProductquantityMS.Text = datagridviewMS.SelectedRows[0].Cells[2].Value.ToString();
            ProductamountMS.Text = datagridviewMS.SelectedRows[0].Cells[3].Value.ToString();

        }

        private void guna2Button3_Click(object sender, EventArgs e)
        {
            if (LoginForm.Userlvl == "guest")
            {
                MessageBox.Show("You are not authorised to make changes");
            }
            else
            {
                con.Open();
                int combocat;
                int myint = 0;
                string query1 = "select * from CategoryTbl where categoryname='" + CategoryMS.SelectedValue.ToString() + "';";
                SqlCommand cmd1 = new SqlCommand(query1, con);
                SqlDataReader rdr;

                rdr = cmd1.ExecuteReader();
                while (rdr.Read())
                {
                    myint = rdr.GetInt32(0);
                }
                rdr.Close();
                combocat = myint;
                SqlCommand cmd = new SqlCommand("update StockTbl set productname='" + ProductnameMS.Text +
"',productquantity='" + ProductquantityMS.Text + "', productamount='" + ProductamountMS.Text + "', productcatid = '"+ combocat
+"' where productid='" + datagridviewMS.SelectedRows[0].Cells[0].Value.ToString() + "';", con);
                cmd.ExecuteNonQuery();
                MessageBox.Show("Product Successfully Updated");
                con.Close();
                populate();
            }
        }

        private void guna2Button5_Click(object sender, EventArgs e)
        {
            filterbycategory();
        }

        private void guna2ComboBox1_SelectedIndexChanged(object sender, EventArgs e)
        {

        }

        private void RefreshMS_Click(object sender, EventArgs e)
        {
            populate();
        }

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void guna2Button4_Click(object sender, EventArgs e)
        {
            if (datagridviewMS.Rows.Count > 0)
            {
                Microsoft.Office.Interop.Excel.Application xcelApp = new Microsoft.Office.Interop.Excel.Application();
                xcelApp.Application.Workbooks.Add(Type.Missing);
                for (int i = 1; i < datagridviewMS.Columns.Count + 1; i++)
                {
                    xcelApp.Cells[1, i] = datagridviewMS.Columns[i - 1].HeaderText;
                }
                for (int i = 0; i < datagridviewMS.Rows.Count; i++)
                {
                    for (int j = 0; j < datagridviewMS.Columns.Count; j++)
                    {
                        xcelApp.Cells[i + 2, j + 1] = datagridviewMS.Rows[i].Cells[j].Value.ToString();
                    }
                }
                xcelApp.Columns.AutoFit();
                xcelApp.Visible = true;
            }
        }

        private void totalprice_TextChanged(object sender, EventArgs e)
        {

        }

        private void ProductamountMS_TextChanged(object sender, EventArgs e)
        {
            double calc;
            try
            {
                double amt = Convert.ToDouble(ProductamountMS.Text);
                double qty = Convert.ToDouble(ProductquantityMS.Text);
                 calc = amt * qty;
                totalprice.Text = calc.ToString();
            }
            catch
            {
                calc = 0.0;
            }

        }
    }
}
```

MANAGE ORDERS :

Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace inventory_management
{
    public partial class Order : Form
    {
        public Order()
        {
            InitializeComponent();
        }
        SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\DEBASISH\Documents\Inventory_
Manage_user.mdf;Integrated Security=True;Connect Timeout=30");
        void populate()
        {
            try
            {
                con.Open();
                string MYquery = "select * from CustomerTbl";
                SqlDataAdapter da = new SqlDataAdapter(MYquery, con);
                SqlCommandBuilder builder = new SqlCommandBuilder(da);
                var ds = new DataSet();
                da.Fill(ds);
                CustomersMO.DataSource = ds.Tables[0];
                con.Close();
            }
            catch (Exception E)
            {
                MessageBox.Show(E.Message);
            }
        }
        void populateproducts()
        {
            try
            {
                con.Open();
                string MYquery = "select productid, productname, productquantity, 
productamount from StockTbl";
                SqlDataAdapter da = new SqlDataAdapter(MYquery, con);
                SqlCommandBuilder builder = new SqlCommandBuilder(da);
                var ds = new DataSet();
                da.Fill(ds);
                ProductMO.DataSource = ds.Tables[0];
                con.Close();
            }
            catch (Exception E)
            {
                MessageBox.Show(E.Message);
            }
        }
        void filterbycategory()
        {
            try
            {
                /* con.Open();
                 string Query = "select categoryid from CategoryTbl where 
categoryname = '" + SearchcomboMO.SelectedValue.ToString() + "' ";
                 int Query1 = Convert.ToInt32( Query);
                 SqlDataAdapter da1 = new SqlDataAdapter(Query, con);
                 SqlCommandBuilder builder1 = new SqlCommandBuilder(da1);
                 string MYquery = "select * from StockTbl where productcatid =  '" + 
Query1 + "' ";
                 SqlDataAdapter da = new SqlDataAdapter(MYquery, con);
                 SqlCommandBuilder builder = new SqlCommandBuilder(da);
                 var ds = new DataSet();
                 da.Fill(ds);
                 ProductMO.DataSource = ds.Tables[0];
                 con.Close(); */
                con.Open();
                int combocat;
                int myint = 0;
                string query1 = "select * from CategoryTbl where categoryname='" + 
SearchcomboMO.SelectedValue.ToString() + "';";
                SqlCommand cmd1 = new SqlCommand(query1, con);
                SqlDataReader rdr= null;

                rdr = cmd1.ExecuteReader();
                while (rdr.Read())
                {
                    myint = rdr.GetInt32(0);
                }
                rdr.Close();
                combocat = myint;
                string MYquery = "select productid, productname, productquantity, 
productamount from StockTbl where productcatid= '" + combocat + "' ";
```

```
                SqlDataAdapter da = new SqlDataAdapter(MYquery, con);
                SqlCommandBuilder builder = new SqlCommandBuilder(da);
                var ds = new DataSet();
                da.Fill(ds);
                ProductMO.DataSource = ds.Tables[0];
                con.Close();
            }
            catch (Exception E)
            {
                MessageBox.Show(E.Message);
                con.Close();
            }
        }
        void Fillcombo()
        {
            string query = "select * from CategoryTbl;";
            SqlCommand cmd = new SqlCommand(query, con);
            SqlDataReader rdr;
            try
            {
                con.Open();
                DataTable dt = new DataTable();
                dt.Columns.Add("categoryname", typeof(string));
                rdr = cmd.ExecuteReader();
                dt.Load(rdr);
                SearchcomboMO.ValueMember = "categoryname";
                SearchcomboMO.DataSource = dt;
                con.Close();
            }
            catch (Exception E)
            {
                MessageBox.Show(E.Message);
            }
        }
        public static int newQty;
        void updateproduct()
        {

            newQty = stock - Convert.ToInt32(QualityMo.Text);
            if (newQty < 0)
            {
                MessageBox.Show("Operation Failed");
            }
            else
            {
               /* con.Open();
                string query = "update StockTbl set productquantity ='" + newQty + "'
where productid = '" + id + "'; ";
                SqlCommand cmd = new SqlCommand(query, con);
                cmd.ExecuteNonQuery();
                con.Close();
                populateproducts();*/
                ProductMO.SelectedRows[0].Cells[2].Value = newQty;
            }
        }
        int num = 0;
        double uprice, totprice;
        int qty;
        string product;
        private void Order_Load(object sender, EventArgs e)
        {

        }
        DataTable table = new DataTable();
        public static int i = 1;
        private void Order_Load_1(object sender, EventArgs e)
        {
            populate();
            populateproducts();
            Fillcombo();
            if(i==1)
            {
                table.Columns.Add("Num", typeof(int));
                table.Columns.Add("Product", typeof(string));
                table.Columns.Add("Quantity", typeof(int));
                table.Columns.Add("PriceperPiece", typeof(double));
                table.Columns.Add("Total Price", typeof(double));
                orderGV.DataSource = table;
                Userlvl.Text = LoginForm.Userlvl;
                i += 1;
            }

        }

        private void dateTimePicker1_ValueChanged(object sender, EventArgs e)
        {

        }

        private void label4_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void BackToHomeMo_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
        {
            Dashboard home = new Dashboard();
            home.Show();
            this.Hide();
        }

        private void guna2DataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {



        }

        private void guna2Button2_Click(object sender, EventArgs e)
        {
            filterbycategory();
        }

        private void guna2Button1_Click(object sender, EventArgs e)
        {
            populateproducts();
        }

        private void CustomersMO_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {
            OrderIdMO.Text = CustomersMO.SelectedRows[0].Cells[0].Value.ToString();

        }
        int flag = 0;
        double sum = 0;
        int stock;
        private void guna2Button3_Click(object sender, EventArgs e)
        {
            if (QualityMo.Text == "")
            {
                MessageBox.Show("Enter The Quantity of Products");
            }
            else if (flag == 0)
            {
                MessageBox.Show("Select The Product");
            }
            else if (Convert.ToInt32(QualityMo.Text) > stock)
                MessageBox.Show("No Enough Stack Available");
            else
            {
                num = num + 1;
                qty = Convert.ToInt32(QualityMo.Text);
                totprice = qty * uprice;
                table.Rows.Add(num, product, qty, uprice, totprice);
                orderGV.DataSource = table;
                updateproduct();
                flag = 0;
                sum = sum + totprice;
                TotAmount.Text = sum.ToString();
            }
        }

        private void QualityMo_TextChanged(object sender, EventArgs e)
        {

        }

        private void InsertButtonMo_Click(object sender, EventArgs e)
        {
            foreach(DataGridViewRow dr in ProductMO.Rows)
            {
                con.Open();
                int id = Convert.ToInt32(dr.Cells[0].Value);
                string query = "update StockTbl set productquantity ='" +
dr.Cells[2].Value + "' where productid = '" + id + "'; ";
                SqlCommand cmd = new SqlCommand(query, con);
                cmd.ExecuteNonQuery();
                con.Close();
            }

            populateproducts();
            if (OrderIdMO.Text == "" ||   TotAmount.Text == "")
            {
                MessageBox.Show("Fill The Data Correctly");
            }
            else
            {

                con.Open();
                SqlCommand cmd2 = new SqlCommand("insert into OrderTbl values('" +
OrderIdMO.Text + "','" + orderdateMO.Text + "','" + TotAmount.Text + "','" + "Not
Paid" + "')", con);
                cmd2.ExecuteNonQuery();
                string myquery = "select * from OrderTbl";
                SqlDataAdapter ad = new SqlDataAdapter(myquery, con);
                DataTable dt = new DataTable();
                ad.Fill(dt);
                con.Close();
                foreach (DataRow dr in table.Rows)
                {
                    con.Open();
                    int id1;
                    int myint = 0;
                    string query1 = "select * from StockTbl where productname='" + 
dr.ItemArray[1] + "';";
                    SqlCommand cmd1 = new SqlCommand(query1, con);
                    SqlDataReader rdr = null;
                    try
                    {
                        rdr = cmd1.ExecuteReader();
                        while (rdr.Read())
                        {
                            myint = rdr.GetInt32(0);
                        }
                        rdr.Close();
                    }
                    catch (Exception E)
                    {
                        MessageBox.Show(E.Message);
                        con.Close();
                    }
                    id1 = myint;
                    SqlCommand cmd3 = new SqlCommand("insert into OrderProductsTbl 
values('" + id1 + "','" + dr.ItemArray[0] + "','" + dr.ItemArray[1] + "','" + 
dr.ItemArray[2] + "','" + dr.ItemArray[3] + "','" + dr.ItemArray[4] + "','" + 
dt.Rows[Convert.ToInt32(dt.Rows.Count) - 1][0] + "')", con);
                    cmd3.ExecuteNonQuery();

                    con.Close();
                }
                MessageBox.Show("Order Successfully Added");
                table.Clear();
                TotAmount.Clear();
                QualityMo.Clear();

                table.Clear();
                con.Close();
            }

        }

        private void ViewButtonMo_Click(object sender, EventArgs e)
        {
            ViewOrders view = new ViewOrders();
            view.Show();
        }

        private void OrderIdMo_TextChanged(object sender, EventArgs e)
        {

        }
        private void SearchcomboMO_SelectedIndexChanged(object sender, EventArgs e)
        {

        }

        private void ProductMO_CellContentClick(object sender, 
DataGridViewCellEventArgs e)
        {
            product = ProductMO.SelectedRows[0].Cells[1].Value.ToString();
           // qty = Convert.ToInt32(QualityMo.Text);
            stock= 
Convert.ToInt32(ProductMO.SelectedRows[0].Cells[2].Value.ToString());
            uprice = 
Convert.ToDouble(ProductMO.SelectedRows[0].Cells[3].Value.ToString());
            //  totprice = qty * uprice;
            flag = 1;
        }
    }
}
```

VIEW ORDERS :

Code:

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;

namespace inventory_management
{
    public partial class ViewOrders : Form
    {
        public ViewOrders()
        {
            InitializeComponent();
        }


        SqlConnection con = new SqlConnection(@"Data 
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\DEBASISH\Documents\Inventory_Mana
ge_user.mdf;Integrated Security=True;Connect Timeout=30");
        void populateorders()
        {
            try
            {
                con.Open();
                string MYquery = "select * from OrderTbl";
                SqlDataAdapter da = new SqlDataAdapter(MYquery, con);
                SqlCommandBuilder builder = new SqlCommandBuilder(da);
                var ds = new DataSet();
                da.Fill(ds);
                OrdersGV.DataSource = ds.Tables[0];
                con.Close();
            }
            catch (Exception E)
            {
                MessageBox.Show(E.Message);
            }
        }
        void Fillcombo()
        {
            string query = "select * from CustomerTbl;";
            SqlCommand cmd = new SqlCommand(query, con);
            SqlDataReader rdr;
            try
            {
                con.Open();
                DataTable dt = new DataTable();
                dt.Columns.Add("customername", typeof(string));
                rdr = cmd.ExecuteReader();
                dt.Load(rdr);
                SearchcomboVO.ValueMember = "customername";
                SearchcomboVO.DataSource = dt;
                con.Close();
            }
            catch (Exception E)
            {
                MessageBox.Show(E.Message);
            }
        }
        void filterbycustomername()
        {
            try
            {

                con.Open();
                int combocat;
                int myint = 0;
                string query1 = "select * from CustomerTbl where customername='" +
SearchcomboVO.Text + "';";
                SqlCommand cmd1 = new SqlCommand(query1, con);
                SqlDataReader rdr = null;

                rdr = cmd1.ExecuteReader();
                while (rdr.Read())
                {
                    myint = rdr.GetInt32(0);
                }
                rdr.Close();
                combocat = myint;
                string MYquery = "select * from OrderTbl where Custid = '" + combocat +
"' ";

                SqlDataAdapter da = new SqlDataAdapter(MYquery, con);
                SqlCommandBuilder builder = new SqlCommandBuilder(da);
                var ds = new DataSet();
                da.Fill(ds);
                OrdersGV.DataSource = ds.Tables[0];
                con.Close();
            }
            catch (Exception E)
            {
                MessageBox.Show(E.Message);
                con.Close();
            }
        }

        void filterbydate()
        {
            try
            {

                con.Open();

                string MYquery = "select * from OrderTbl where OrderDate = '" +
orderdateVO.Text + "';";
                SqlDataAdapter da = new SqlDataAdapter(MYquery, con);
                SqlCommandBuilder builder = new SqlCommandBuilder(da);
                var ds = new DataSet();
                da.Fill(ds);
                OrdersGV.DataSource = ds.Tables[0];
                con.Close();
            }
            catch (Exception E)
            {
                MessageBox.Show(E.Message);
                con.Close();
            }
        }
        private void label4_Click(object sender, EventArgs e)
        {

        }

        private void panel2_Paint(object sender, PaintEventArgs e)
        {

        }

        private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {
            /* if(printPreviewDialog1.ShowDialog() == DialogResult.OK)
             {
                 printDocument1.Print();
             }
             printDocument1.Print();*/
        }

        private void DeleteButtonMc_Click(object sender, EventArgs e)
        {
            this.Hide();
        }

        private void ViewOrders_Load(object sender, EventArgs e)
        {
            populateorders();
            Fillcombo();
        }

        private void guna2Button3_Click(object sender, EventArgs e)
        {
            if (OrdersGV.Rows.Count > 0)
            {
                Microsoft.Office.Interop.Excel.Application xcelApp = new
Microsoft.Office.Interop.Excel.Application();
                xcelApp.Application.Workbooks.Add(Type.Missing);
                for (int i = 1; i < OrdersGV.Columns.Count + 1; i++)
                {
                    xcelApp.Cells[1, i] = OrdersGV.Columns[i - 1].HeaderText;
                }
                for (int i = 0; i < OrdersGV.Rows.Count; i++)
                {
                    for (int j = 0; j < OrdersGV.Columns.Count; j++)
                    {
                        xcelApp.Cells[i + 2, j + 1] =
OrdersGV.Rows[i].Cells[j].Value.ToString();
                    }
                }
                xcelApp.Columns.AutoFit();
                xcelApp.Visible = true;
            }
        }

        private void printDocument1_PrintPage(object sender,
System.Drawing.Printing.PrintPageEventArgs e)
        {
            //  e.Graphics.DrawString("Order Summary", new Font("Century", 25,
FontStyle.Bold), Brushes.Red,new Point(230));
            //e.Graphics.DrawString("Order Id:" +
OrdersGV.SelectedRows[0].Cells[0].Value.ToString(), new Font("Century", 25,
FontStyle.Italic), Brushes.Black, new Point(80, 100));

        }
        public static int orderid;
        public static int customerid;
        public static string status = "";
        private void OrdersGV_CellContentClick(object sender, DataGridViewCellEventArgs
e)
        {
            orderid = Convert.ToInt32(OrdersGV.SelectedRows[0].Cells[0].Value);
            customerid = Convert.ToInt32(OrdersGV.SelectedRows[0].Cells[1].Value);
            status =  Convert.ToString(OrdersGV.SelectedRows[0].Cells[4].Value);

            OrderProducts db = new OrderProducts();
            db.Visible = true;
        }

        private void guna2Button1_Click(object sender, EventArgs e)
        {
            filterbycustomername();
        }

        private void SearchcomboMO_SelectedIndexChanged(object sender, EventArgs e)
        {

        }

        private void orderdateMO_ValueChanged(object sender, EventArgs e)
        {

        }
```

```
            private void guna2Button2_Click(object sender, EventArgs e)
            {
                filterbydate();
            }

            private void guna2Button4_Click(object sender, EventArgs e)
            {
                populateorders();
            }
        }
}
```

VIEW PRODUCTS :

Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.IO;
using System.Diagnostics;

namespace inventory_management
{
    public partial class OrderProducts : Form
    {
        public OrderProducts()
        {
            InitializeComponent();
        }
        SqlConnection con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\DEBASISH\Documents\Inventory_Manage_user.mdf;Integrated Security=True;Connect Timeout=30");
        void populateproducts()
        {
            try
            {
                con.Open();
                string MYquery = "select num,productid,quantity,productprpiece,totamt from OrderProductsTbl where orderid = '"+ViewOrders.orderid+"' ";
                SqlDataAdapter da = new SqlDataAdapter(MYquery, con);
                SqlCommandBuilder builder = new SqlCommandBuilder(da);
                var ds = new DataSet();
                da.Fill(ds);
                OrdersProductsGV.DataSource = ds.Tables[0];
                con.Close();
            }
            catch (Exception E)
            {
                MessageBox.Show(E.Message);
            }
        }
        private void OrderProducts_Load(object sender, EventArgs e)
        {
            Status.Text = ViewOrders.status;
            if(Status.Text == "Paid")
            {
                Status.ForeColor = Color.Green;
            }
            else
            {
                Status.ForeColor = Color.Red;
            }
            OrderId.Text ="OrderId : " + Convert.ToString(ViewOrders.orderid);
            CustomerId.Text ="CustomerId : " + Convert.ToString(ViewOrders.customerid);
            populateproducts();
        }

        private void guna2Button2_Click(object sender, EventArgs e)
        {
            if (OrdersProductsGV.Rows.Count > 0)
            {
                Microsoft.Office.Interop.Excel.Application xcelApp = new Microsoft.Office.Interop.Excel.Application();
                xcelApp.Application.Workbooks.Add(Type.Missing);
                for (int i = 1; i < OrdersProductsGV.Columns.Count + 1; i++)
                {
                    xcelApp.Cells[1, i] = OrdersProductsGV.Columns[i - 1].HeaderText;
                }
                for (int i = 0; i < OrdersProductsGV.Rows.Count; i++)
                {
                    for (int j = 0; j < OrdersProductsGV.Columns.Count; j++)
                    {
                        xcelApp.Cells[i + 2, j + 1] = OrdersProductsGV.Rows[i].Cells[j].Value.ToString();
                    }
                }
                xcelApp.Columns.AutoFit();
                xcelApp.Visible = true;
            }
        }

        private void DeleteButtonMc_Click(object sender, EventArgs e)
        {
            if(Status.Text == "Paid")
            {
```

```
                con.Open();
                SqlCommand cmd = new SqlCommand("update OrderTbl set Status='" + Status.Text + "' where Orderid= '" +
ViewOrders.orderid + "';", con);
                MessageBox.Show("Order Paid");
                cmd.ExecuteNonQuery();
                con.Close();

            }
            this.Visible = false;
        }

        public static string ProdName;
        public static string prodname;

        private void printDocument1_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)
        {
            e.Graphics.DrawString("Order Summary", new System.Drawing.Font("California FB", 30, FontStyle.Bold),
Brushes.Red, new System.Drawing.Point(150));
            e.Graphics.DrawString(OrderId.Text, new System.Drawing.Font("Cambria", 20, FontStyle.Regular), Brushes.Green,
new System.Drawing.Point(70, 50));
            e.Graphics.DrawString(CustomerId.Text, new System.Drawing.Font("Cambria", 20, FontStyle.Regular),
Brushes.Green, new System.Drawing.Point(70, 90));
            con.Open();
            string squery = "select * from OrderTbl where Orderid = '" + ViewOrders.orderid + "';";

            SqlCommand CMD = new SqlCommand(squery, con);
            SqlDataReader rdr = null;
            string orderdate = "";
            string totalamt = "";
            try
            {
                rdr = CMD.ExecuteReader();
                while (rdr.Read())
                {
                    orderdate = Convert.ToString(rdr.GetDateTime(2));
                    totalamt = Convert.ToString(rdr.GetValue(3));
                }
                rdr.Close();
            }
            catch (Exception E)
            {
                MessageBox.Show(E.Message);
                con.Close();
            }
            con.Close();
            string id1 = orderdate;
            string id2 = totalamt;
            e.Graphics.DrawString("Order Date: " + id1, new Font("Cambria", 20, FontStyle.Regular), Brushes.Green, new
Point(70, 140));
```

```
e.Graphics.DrawString("Order Total Amount: " + id2, new Font("Cambria", 20, FontStyle.Regular), Brushes.Green, new Point(70, 190));
string[] Arr = { "Sl.No.", "ProductName", "Product Qty.", "PricePerUnit", "TotalPrice" };
int x = 20;
int g = 1;
con.Open();
string myquery = "select * from OrderTbl";
SqlDataAdapter da = new SqlDataAdapter(myquery, con);
DataTable Table = new DataTable();
da.Fill(Table);
con.Close();
foreach (string str in Arr)
{

    e.Graphics.DrawString(str, new Font("Californian FB", 18, FontStyle.Underline), Brushes.SandyBrown, new Point(x, 245));
    if (g == 2)
    {
        x += 190;
    }
    else
    {
        x += 150;
    }
    g += 1;
}
int j = 265;
con.Open();
string query = "select * from OrderProductsTbl where orderid = '" + ViewOrders.orderid + "'";
SqlDataAdapter daa = new SqlDataAdapter(query, con);
DataTable Table2 = new DataTable();
daa.Fill(Table2);
con.Close();
foreach (DataRow dr in Table2.Rows)
{
    int i = 20;
    int gap = 1;
    //MessageBox.Show("Got here!");
    j += 30;
    foreach (var item in dr.ItemArray)
    {
        if (gap == 2)
        {
            con.Open();
            string Query = "select * from StockTbl where productid = '" + Convert.ToInt32(item) + "';";

            SqlCommand Cmd = new SqlCommand(Query, con);
            SqlDataReader reader = null;

            try
```

```
                        {
                            reader = Cmd.ExecuteReader();
                            while (reader.Read())
                            {
                                prodname = reader.GetString(1);
                            }
                            reader.Close();
                        }
                        catch (Exception E)
                        {
                            MessageBox.Show(E.Message);
                            con.Close();
                        }
                        con.Close();
                        ProdName = prodname;
                        e.Graphics.DrawString(ProdName, new Font("Californian FB", 17, FontStyle.Bold), Brushes.Black, new
Point(i, j));
                        i += 190;
                    }
                    else if (gap >= 3)
                    {
                        e.Graphics.DrawString(item.ToString(), new Font("Californian FB", 17, FontStyle.Bold), Brushes.Black,
new Point(i, j));

                        i += 160;

                    }
                    //MessageBox.Show("Got here!");

                    gap += 1;
                }

            }
        }

        private void guna2Button3_Click(object sender, EventArgs e)
        {
            Status.Text = "Paid";
            Status.ForeColor = Color.Green;

            printDocument1.Print();

        }
        private void CustomerId_Click(object sender, EventArgs e)
        {

        }

        private void label2_Click(object sender, EventArgs e)
        {
        }

        private void label2_Click_1(object sender, EventArgs e)
        {

        }
    }
}
```

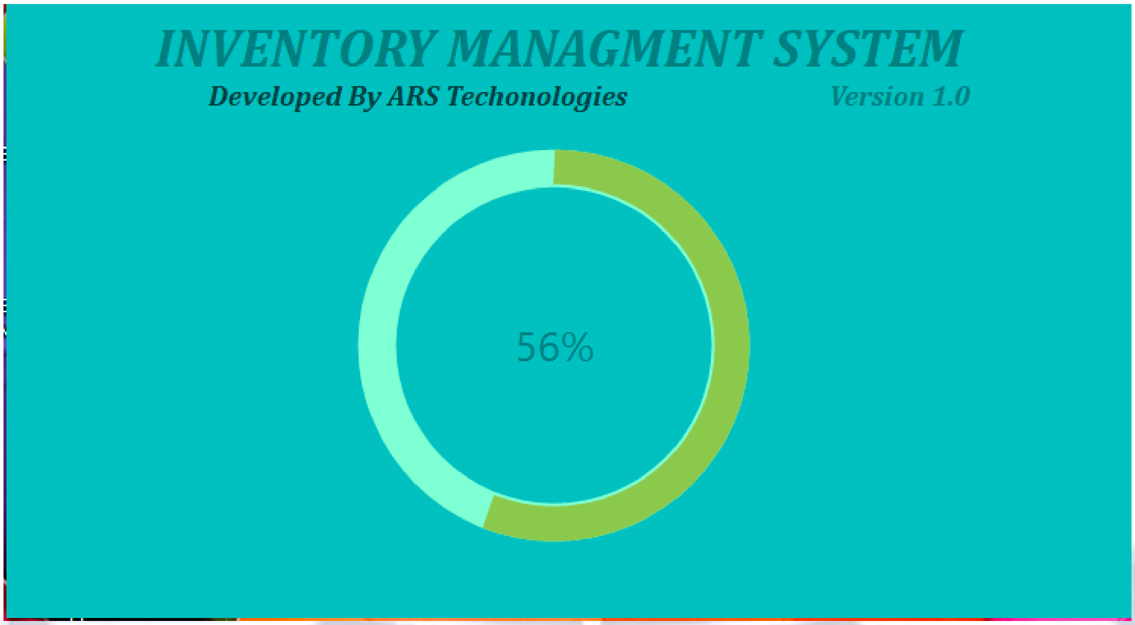
MANAGE SUPPLIERS :

Code :

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
namespace inventory_management
{
    public partial class supplier : Form
    {
        public supplier()
        {
            InitializeComponent();
        }
        SqlConnection con = new SqlConnection(@"Data 
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\DEBASISH\Documents\Inventory_Mana
ge_user.mdf;Integrated Security=True;Connect Timeout=30");
        void populate()
        {
            try
            {
                con.Open();
                string MYquery = "select 
supplierid,suppliername,supplierno,supplieraddress from SupplierTbl";
                SqlDataAdapter da = new SqlDataAdapter(MYquery, con);
                SqlCommandBuilder builder = new SqlCommandBuilder(da);
                var ds = new DataSet();
                da.Fill(ds);
                datagridviewMS.DataSource = ds.Tables[0];
                con.Close();
            }
            catch (Exception E)
            {
                MessageBox.Show(E.Message);
            }
        }
        private void label4_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs
e)
        {

            Dashboard home = new Dashboard();
            home.Show();
            this.Hide();
        }

        private void guna2Button1_Click(object sender, EventArgs e)
        {
            try
            {
                if ( SuppliernameMS.Text == "" || SuppliernumberMS.Text == "" ||
SupplieraddressMS.Text == "")
                {
                    MessageBox.Show(" Fill all the fields");
                }
                else
                {
                    con.Open();
                    SqlCommand cmd = new SqlCommand("insert into
SupplierTbl(suppliername, supplierno, supplieraddress) values('" + SuppliernameMS.Text +
"','" + SuppliernumberMS.Text + "','" + SupplieraddressMS.Text + "')", con);
                    cmd.ExecuteNonQuery();
                    MessageBox.Show("Supplier Successfully Added");
                    con.Close();
                }
                populate();
            }
            catch (Exception E)
            {
                MessageBox.Show(E.Message);
                MessageBox.Show("Please ensure if the supplier Name nd Supplier Phone no.
are unique!");
                con.Close();

            }
        }

        private void guna2Button2_Click(object sender, EventArgs e)
        {
            try
            {


                if (SuppliernameMS.Text == "")
                {
                    MessageBox.Show("Please Select a supplier");

                }
                else
                {
                    con.Open();
                    string myquery = "delete from SupplierTbl where supplierid='" +
datagridviewMS.SelectedRows[0].Cells[0].Value.ToString() + "'; ";
                    SqlCommand cmd = new SqlCommand(myquery, con);
                    cmd.ExecuteNonQuery();
                    MessageBox.Show("Supplier Successfully Deleted");
                    con.Close();
                    populate();
                }
            }
            catch (Exception E)
            {
                MessageBox.Show(E.Message);
            }
        }

        private void datagridviewMS_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {

            SuppliernameMS.Text =
datagridviewMS.SelectedRows[0].Cells[1].Value.ToString();
            SuppliernumberMS.Text =
datagridviewMS.SelectedRows[0].Cells[2].Value.ToString();
            SupplieraddressMS.Text =
datagridviewMS.SelectedRows[0].Cells[3].Value.ToString();
        }

        private void guna2Button3_Click(object sender, EventArgs e)
        {
            try
            {
                con.Open();
                SqlCommand cmd = new SqlCommand("update SupplierTbl set suppliername='" +
SuppliernameMS.Text + "',supplierno='" + SuppliernumberMS.Text + "', supplieraddress='" +
SupplieraddressMS.Text + "' where supplierid='" +
datagridviewMS.SelectedRows[0].Cells[0].Value.ToString() + "'", con);
                cmd.ExecuteNonQuery();
                MessageBox.Show("Supplier Successfully Updated");
                con.Close();
                populate();
            }
            catch(Exception E)
            {
                MessageBox.Show(E.Message);
                MessageBox.Show("Please ensure if the Supplier Name and Supplier Phone
no. are unique!");
                con.Close();
            }
        }

        private void supplier_Load(object sender, EventArgs e)
        {
            populate();
            label10.Text = LoginForm.Userlvl;
        }

        private void guna2Button4_Click(object sender, EventArgs e)
        {
            if (datagridviewMS.Rows.Count > 0)
            {
                Microsoft.Office.Interop.Excel.Application xcelApp = new
Microsoft.Office.Interop.Excel.Application();
                xcelApp.Application.Workbooks.Add(Type.Missing);
                for (int i = 1; i < datagridviewMS.Columns.Count + 1; i++)
                {
                    xcelApp.Cells[1, i] = datagridviewMS.Columns[i - 1].HeaderText;
                }
                for (int i = 0; i < datagridviewMS.Rows.Count; i++)
                {
                    for (int j = 0; j < datagridviewMS.Columns.Count; j++)
                    {
                        xcelApp.Cells[i + 2, j + 1] = 
datagridviewMS.Rows[i].Cells[j].Value.ToString();
                    }
                }
                xcelApp.Columns.AutoFit();
                xcelApp.Visible = true;
            }
        }
    }
    }
```

RESULT

SPLASH FORM

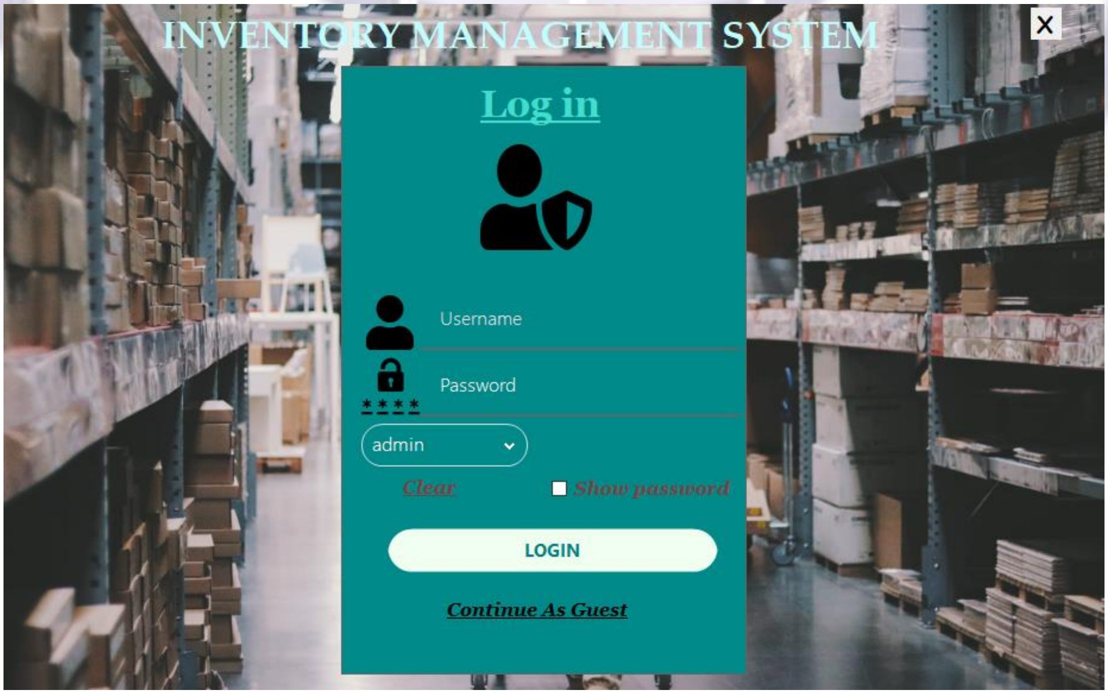


INVENTORY MANAGMENT SYSTEM

Developed By ARS Techonologies

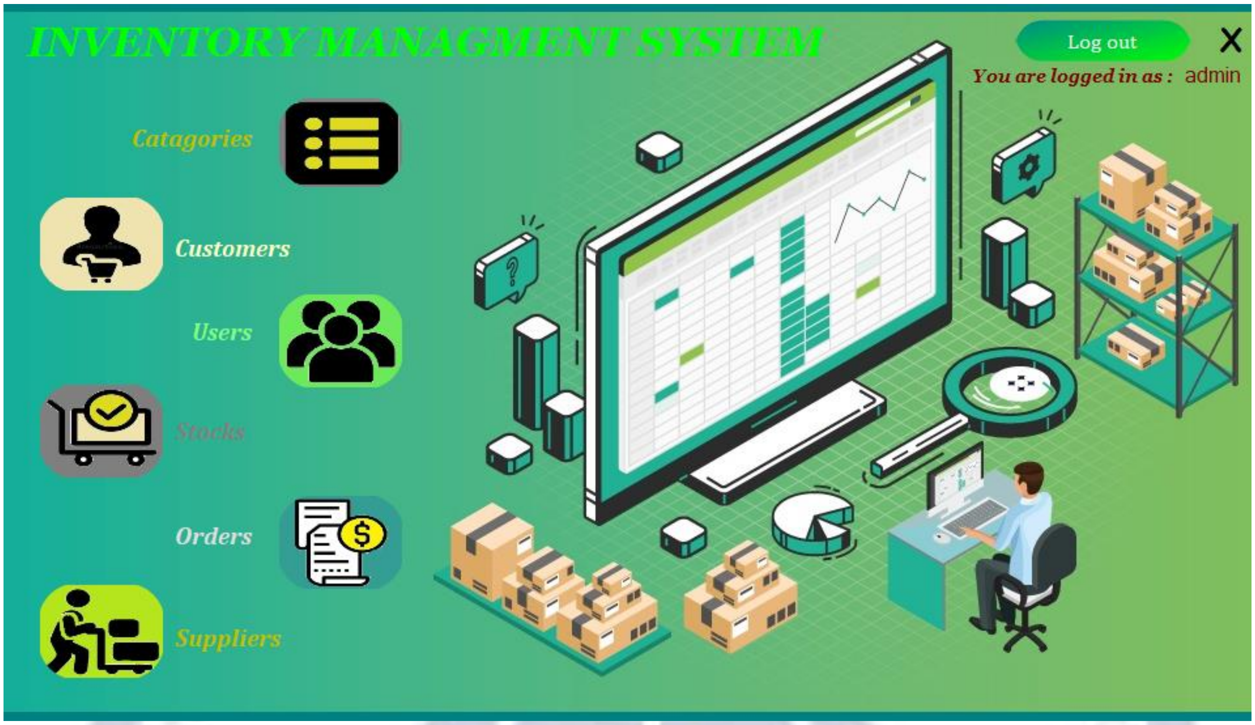
Version 1.0

LOG-IN FORM

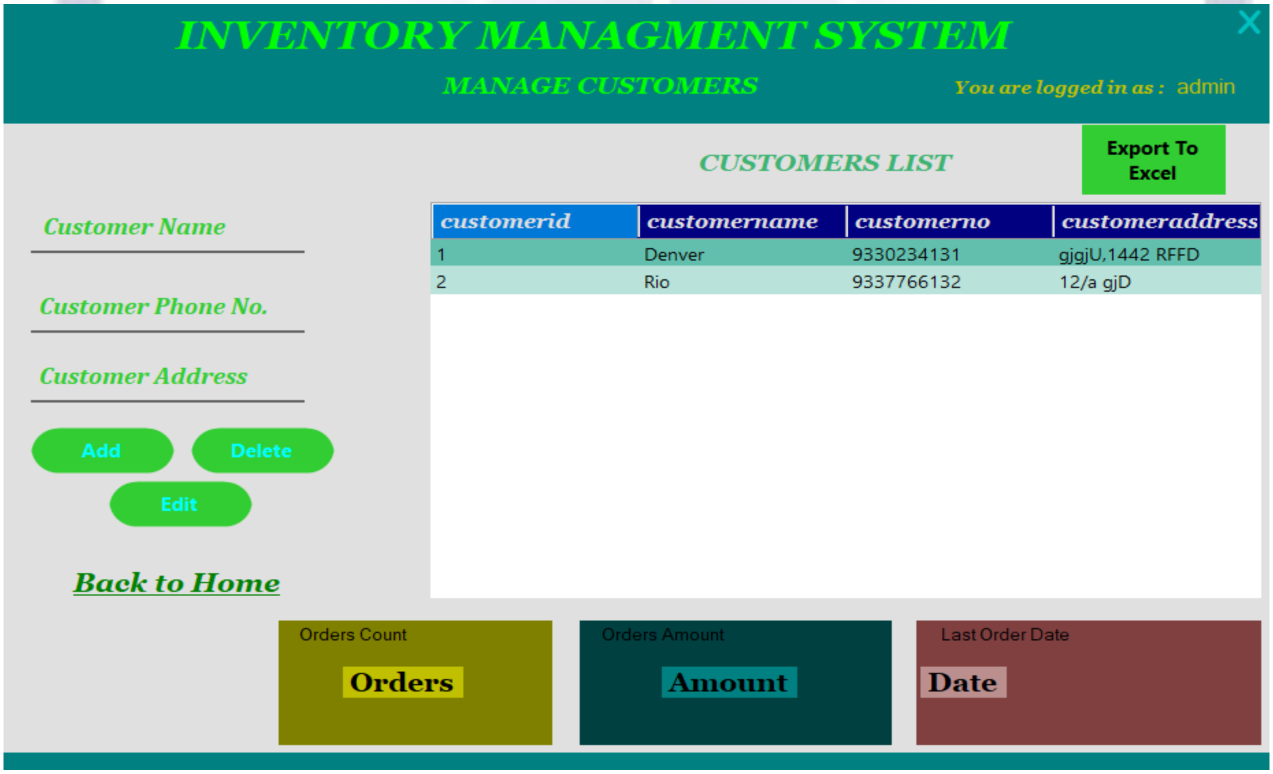


INVENTORY MANAGEMENT SYSTEM

DASHBOARD

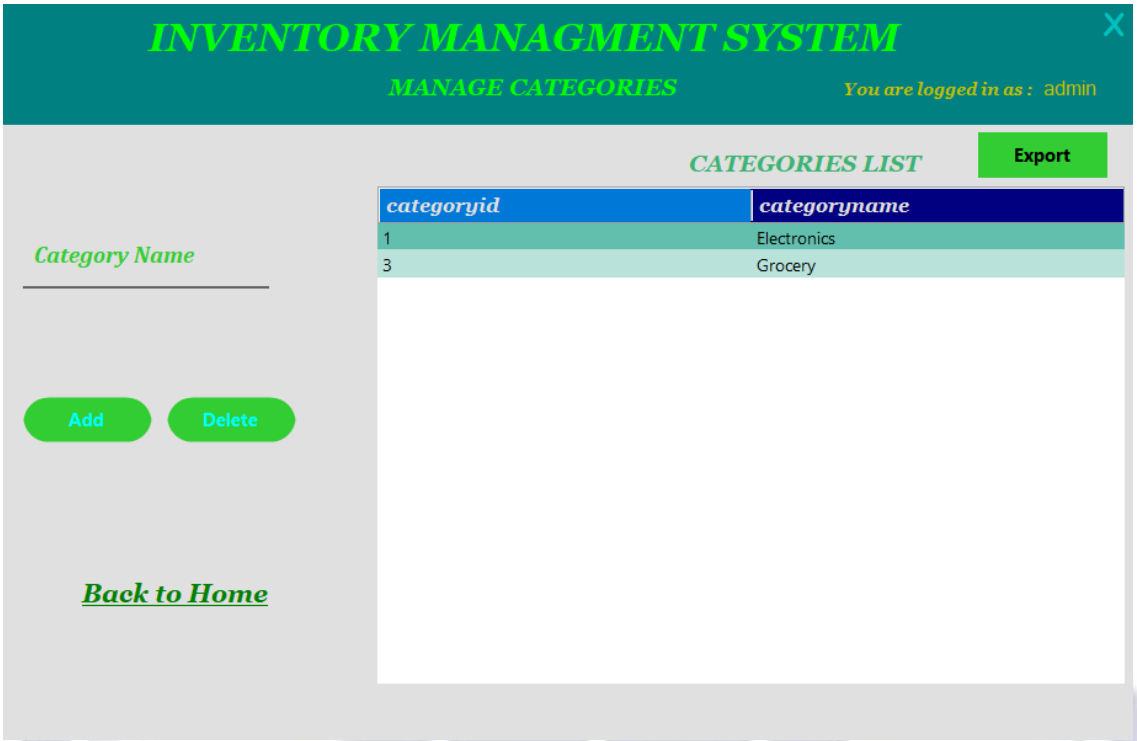


MANAGE CUSTOMERS



customerid	customername	customerno	customeraddress
1	Denver	9330234131	gjgjU,1442 RFFD
2	Rio	9337766132	12/a gjD

MANAGE CATEGORIES



INVENTORY MANAGMENT SYSTEM

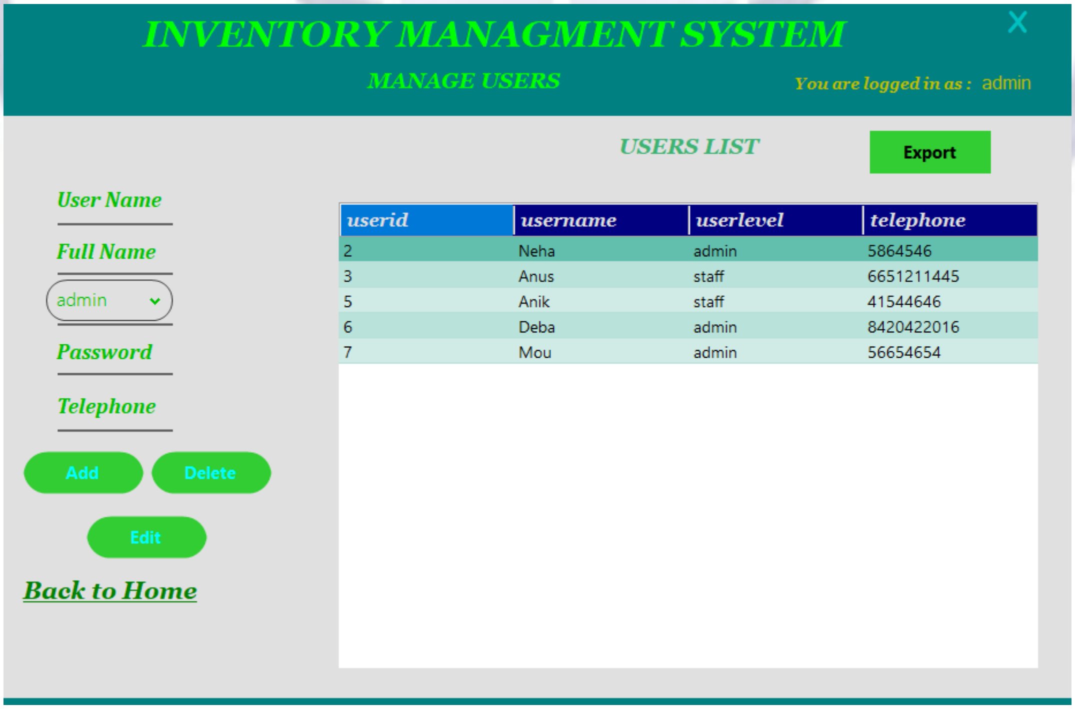
MANAGE CATEGORIES

You are logged in as : admin

CATEGORIES LIST

categoryid	categoryname
1	Electronics
3	Grocery

MANAGE USERS



INVENTORY MANAGMENT SYSTEM

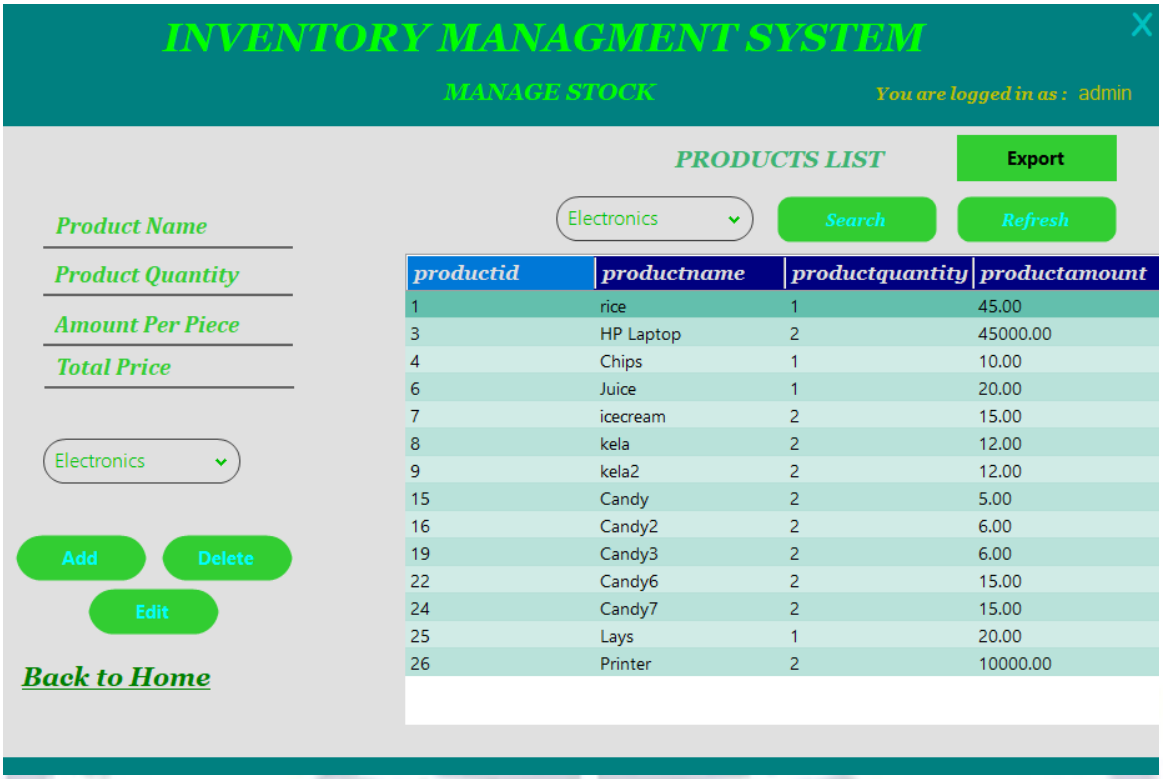
MANAGE USERS

You are logged in as : admin

USERS LIST

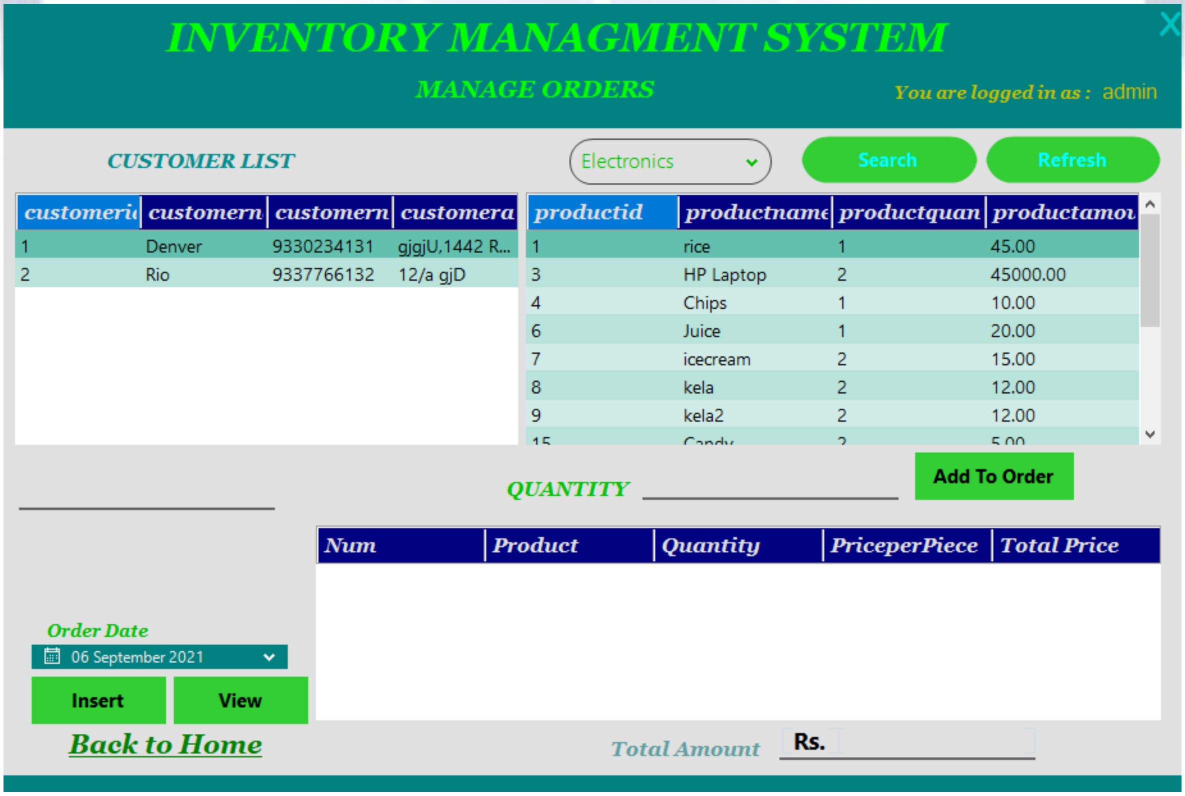
userid	username	userlevel	telephone
2	Neha	admin	5864546
3	Anus	staff	6651211445
5	Anik	staff	41544646
6	Deba	admin	8420422016
7	Mou	admin	56654654

MANAGE STOCK



productid	productname	productquantity	productamount
1	rice	1	45.00
3	HP Laptop	2	45000.00
4	Chips	1	10.00
6	Juice	1	20.00
7	icecream	2	15.00
8	kela	2	12.00
9	kela2	2	12.00
15	Candy	2	5.00
16	Candy2	2	6.00
19	Candy3	2	6.00
22	Candy6	2	15.00
24	Candy7	2	15.00
25	Lays	1	20.00
26	Printer	2	10000.00

MANAGE ORDERS



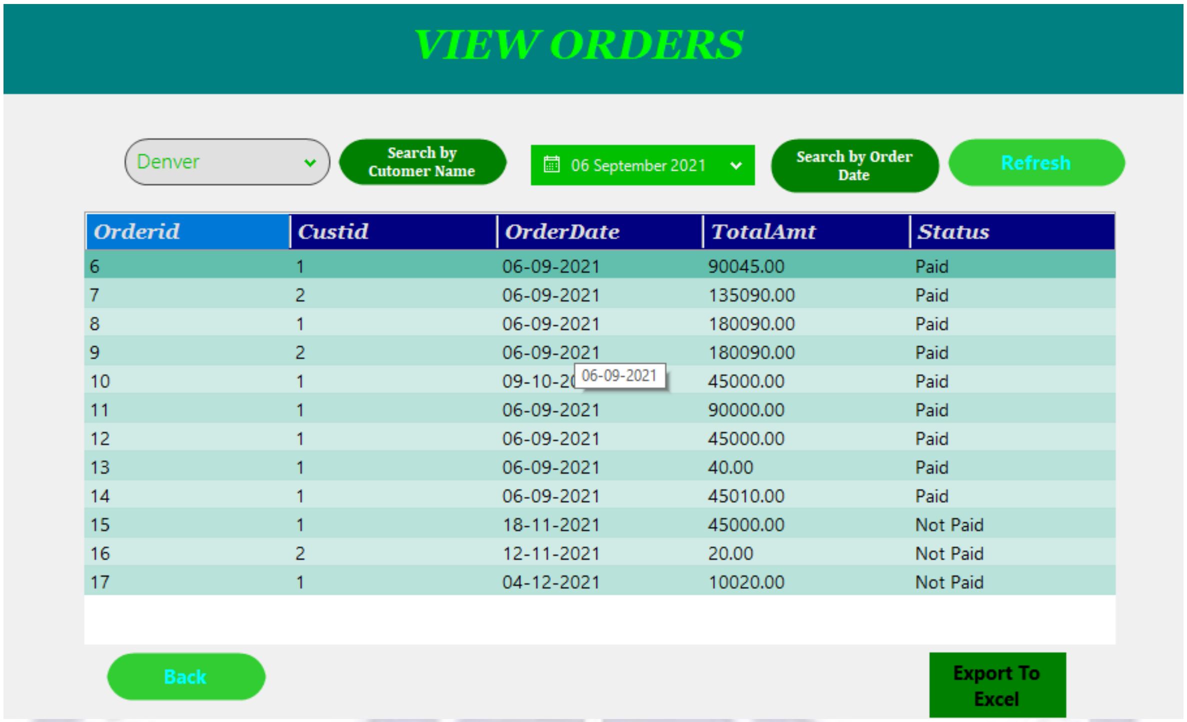
customeri	customern	customern	customera
1	Denver	9330234131	gjgjU,1442 R...
2	Rio	9337766132	12/a gjD

productid	productname	productquan	productamou
1	rice	1	45.00
3	HP Laptop	2	45000.00
4	Chips	1	10.00
6	Juice	1	20.00
7	icecream	2	15.00
8	kela	2	12.00
9	kela2	2	12.00
15	Candy	2	5.00

Num	Product	Quantity	PriceperPiece	Total Price

Order Date: 06 September 2021

VIEW ORDERS

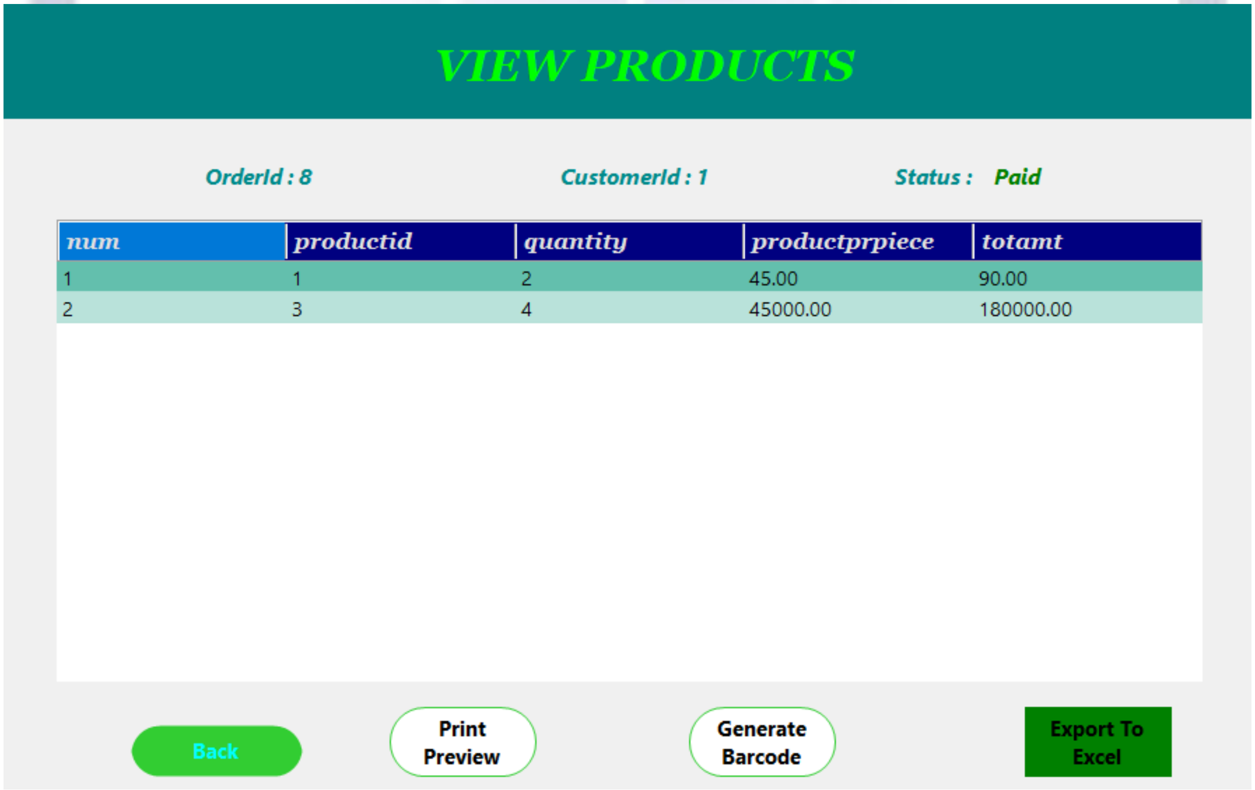


Denver

06 September 2021

Orderid	Custid	OrderDate	TotalAmt	Status
6	1	06-09-2021	90045.00	Paid
7	2	06-09-2021	135090.00	Paid
8	1	06-09-2021	180090.00	Paid
9	2	06-09-2021	180090.00	Paid
10	1	09-10-20	45000.00	Paid
11	1	06-09-2021	90000.00	Paid
12	1	06-09-2021	45000.00	Paid
13	1	06-09-2021	40.00	Paid
14	1	06-09-2021	45010.00	Paid
15	1	18-11-2021	45000.00	Not Paid
16	2	12-11-2021	20.00	Not Paid
17	1	04-12-2021	10020.00	Not Paid

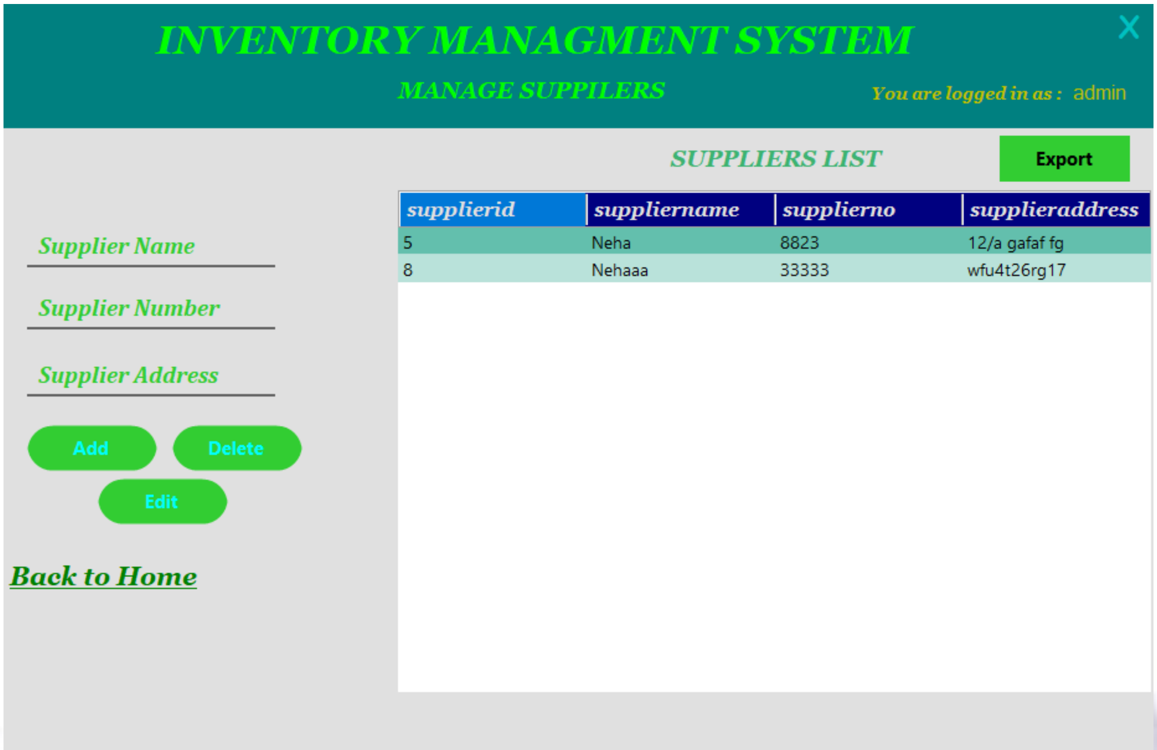
VIEW PRODUCTS



OrderId : 8 CustomerId : 1 Status : Paid

num	productid	quantity	productprpiece	totamt
1	1	2	45.00	90.00
2	3	4	45000.00	180000.00

MANAGE SUPPLIERS



supplierid	suppliername	supplierno	supplieraddress
5	Neha	8823	12/a gafaf fg
8	Nehaaa	33333	wfu4t26rg17

CONCLUSION

Inventory management has to do with keeping accurate records of goods that are ready for shipment. This often means having enough stock of goods to the inventory totals as well as subtracting the most recent shipments of finished goods to buyers. When the company has a return policy in place, there is usually a sub-category contained in the finished goods inventory to account for any returned goods that are reclassified or second grade quality. Accurately maintaining figures on the finished goods inventory makes it possible to quickly convey information to sales personnel as to what is available and ready for shipment at any given time by buyer. Inventory management is important for keeping costs down, while meeting regulation. Supply and demand is a delicate balance, and inventory management hopes to ensure that the balance is undisturbed. Highly trained Inventory management and high-quality software will help make Inventory management a success.

SCOPE FOR FUTURE WORK

The scope of an inventory system can cover many needs, including valuing the inventory, measuring the change in inventory and planning for future inventory levels. The value of the inventory at the end of each period provides a basis for financial reporting on the balance sheet. Measuring the change in inventory allows the company to determine the cost of inventory sold during the period. This allows the company to plan for future inventory needs. Moreover, a threshold point must declare in the stock for which if any product quantity gets below the threshold point then a message will automatically generate to inform that the product must be purchased. The online payment system can also be attached in the “Manage Orders” so that the customer can make online payment then and there.

REFERENCES

1. D. Dhoka, Y.L. Choudary “ABC Classification for Inventory Optimization,” IOSR Journal of Business and Management
2. D.C.U. Cadavid, C.C. Zuluaga, “A framework for decision support system in inventory management area,” Ninth LACCEI Latin American and Caribbean Conf., LACCEI’2011
3. Database System Concepts by Abraham Silberschatz and S Sudarshan
4. C# in Depth by John Skeet
5. Visual Studio 2019 tutorials
6. MySQL server management studio tutorials
7. Various other open sources from Google and YouTube